

# FlexPro 10

A new dimension of

## Data Analysis & Presentation

Quickly and easily organize, analyze and present data





# Contents

<b>1</b>	<b>Introduction .....</b>	<b>9</b>
1.1	Welcome to FlexPro 10 .....	9
1.2	Comparison of FlexPro Editions .....	9
1.3	Installation .....	11
1.4	License Manager .....	13
1.5	Working with the License Manager .....	14
	Opening the License Manager .....	14
	Online Activation .....	15
	Offline Activation .....	15
	Checking Out a Network License .....	16
	Returning a Checked Out License Early .....	17
	Transferring a License .....	17
	Troubleshooting .....	18
<b>2</b>	<b>Tutorials .....</b>	<b>20</b>
2.1	For Newcomers .....	20
	Getting to Know FlexPro in just 15 Minutes .....	20
	Tips for New Users .....	20
2.2	Data Analysis .....	32
2.3	Data Analysis .....	32
	Test Series Analysis Tutorial .....	32
2.4	Customizing FlexPro .....	49
2.5	Customizing FlexPro .....	49
<b>3</b>	<b>Data Management .....</b>	<b>50</b>
3.1	Project Database .....	50
	Working with Project Databases .....	53
	Exporting Project Databases as a Web in HTML Format .....	71
3.2	FlexPro Main Window .....	75
	Working with Windows .....	78
	Working with the Object List .....	79
	Working with the Object Hierarchy .....	84
	Working with the Properties Window .....	85
	Working with the Event Log .....	87
3.3	FlexPro Objects .....	87
	Working with Objects .....	90

3.4	Data Objects .....	98
	Working with Data Objects .....	107
3.5	Data Sets .....	110
	Working with Data Sets .....	113
	Working with Data .....	116
3.6	Excel Workbook .....	122
	Excel Data Link .....	123
	Working with Excel Workbooks .....	123
	Working with Excel Data Links .....	125
3.7	Unit Manager .....	127
	Working with the Unit Manager .....	133
3.8	Importing Files .....	138
	Importing Data .....	141
3.9	Data Export .....	148
	Exporting Data .....	153
3.10	Data Explorer Option .....	154
	Using Data Explorer .....	156
3.11	Data Queries .....	158
	Working with Data Queries .....	159
<b>4</b>	<b>Analyzing Data with Cursors .....</b>	<b>161</b>
4.1	Toggling Cursors .....	167
	Toggling Cursors in a Document or Diagram .....	167
	Toggling Harmonic Cursors .....	167
	Changing the Number and Spacing of Harmonic Cursors .....	167
	Excluding a Curve from Cursors .....	167
4.2	Moving the Cursors .....	168
	Moving the Cursors Using the Keyboard .....	168
	Moving the Cursors with the Mouse .....	168
	Navigating to Peak Values or Markers .....	169
	Navigating to a Specific Point in a Diagram .....	170
	Navigating to a Frame in a Video .....	170
	Scrolling along a Curve .....	170
	Synchronizing Cursors .....	171
4.3	Changing the Image Section .....	171
	Stretching the Image Section .....	171
	Synchronizing Zoom Actions .....	171
	Stretching and Compressing the Curve .....	172
	Using the Mouse to Zoom .....	172
	Scrolling .....	172

---

	Adapting the Image Section to the Curve .....	173
	Adjusting the Image Section when Moving Cursors .....	173
	Undoing Zoom Actions .....	173
	Automatic Zooming of Segments .....	174
4.4	Working with Markers .....	174
	Setting and Deleting Markers .....	174
	Positioning Markers .....	176
	Assigning Media as a Source for Image Markers .....	176
	Editing Marker Text .....	176
	Changing Marker Size .....	178
	Changing Marker Formatting .....	179
	Customizing the Display of Markers .....	180
	Copying Markers to the Clipboard .....	182
4.5	Working with Data .....	182
	Editing Data .....	182
	Setting Data to Void .....	183
	Interpolating Data .....	183
	Copying Data to the Clipboard .....	184
	Displaying Data of a Curve .....	184
4.6	Working with the Coordinates Window .....	184
	Displaying and Hiding the Coordinates Window .....	184
	Changing Coordinates Window Display Options .....	185
	Changing the Coordinates Window Output Format .....	185
	Opening a Data Set Displayed in the Coordinates Window .....	185
	Copying a Value from the Coordinates Window .....	185
	Copying the Contents of the Coordinates Window to the Clipboard .....	186
	Inserting Dynamic Coordinates into a Diagram, Document or Worksheet .....	186
	Adding a Custom Coordinate .....	187
	Editing a Custom Coordinate .....	187
	Displaying and Hiding a Custom Coordinate .....	187
	Displaying a Presentation Object in the Coordinates Window .....	188
	Dimensioning Curves .....	189
	Translating Curves .....	189
	Canceling a Curve Translation .....	190
<b>5</b>	<b>Analyzing Data Mathematically .....</b>	<b>191</b>
5.1	Working with Analyses .....	194
	Structuring Data Analyses .....	194
	Analyzing Multiple Measurements .....	196
	Efficient Data Analysis .....	197

---

5.2	Working with Analysis Templates .....	198
	Creating an Analysis Template .....	198
	Using an Analysis Template .....	198
	Organizing Analysis Templates .....	199
5.3	Formulas .....	200
	Working with the Formula Debugger .....	201
	Working with Formulas .....	205
	FPScript .....	207
5.4	FPScript Functions .....	241
	Working with FPScript Functions .....	242
5.5	Analysis Objects .....	244
	Working with Analysis Objects .....	245
5.6	Reference .....	246
	Analysis Objects and Templates .....	246
	FPScript Operators .....	251
	FPScript Statements .....	259
	FPScript Keywords .....	261
	FPScript Keywords .....	261
	FPScript Properties .....	263
	FPScript Properties .....	263
	FPScript Functions .....	268
<b>6</b>	<b>Presenting Data .....</b>	<b>294</b>
6.1	2D and 3D Diagrams .....	297
	Axis .....	298
	Curve .....	302
	2D Curve Display Formats .....	303
	3D Curve Types and Display Formats .....	305
	Legend .....	309
	Axis and Curve Labeling .....	310
	3D View and Lighting .....	310
	Polar Graph .....	312
	Color Palette and Color Gradient .....	314
	Grid .....	315
	Color Legend .....	315
	Working with Diagrams .....	315
	Working with Axes .....	320
	Working with Curves .....	329
	Working with Legends .....	345
6.2	Column and Cell Tables .....	347

---

Working with Tables .....	349
Working with Column Tables .....	352
Working with Cell Tables .....	357
6.3 Text .....	360
Working with Text .....	360
6.4 Media .....	364
Working with Media .....	366
6.5 Documents .....	369
Working with Documents .....	371
Working with Drawings and Other Objects .....	377
6.6 Drawing Objects .....	378
Working with Drawing Objects .....	379
6.7 Worksheets .....	383
Working with Worksheets .....	384
6.8 Exporting Presentation Objects, Documents and Worksheets .....	387
Exporting Presentation Objects, Documents and Worksheets as a File .....	387
Exporting Presentation Objects, Documents and Worksheets with OLE .....	388
6.9 Working with Presentation Templates .....	389
Creating a Presentation or Document Template .....	389
Using a Presentation or Document Template .....	389
Organizing Presentation and Document Templates .....	390
6.10 Hyperlinks .....	391
Working with Hyperlinks .....	391
6.11 Formatter .....	393
6.12 Field and Placeholder .....	399
<b>7 Customizing FlexPro .....</b>	<b>403</b>
7.1 Customizing the User Interface .....	403
Customizing Menus and Toolbars .....	404
Customizing Shortcuts .....	420
7.2 User Profile .....	421
Working with User Profiles and User Rights .....	421
7.3 Using Startup Switches to Launch FlexPro .....	424
7.4 Customizing the FlexPro Undo Function .....	425
7.5 Customizing FlexPro Messages .....	425
7.6 Customizing the Unit for Dimension Measures .....	425
7.7 Customizing the Decimal Symbol .....	425
7.8 Adjusting the Output Format for Numeric Data .....	426
7.9 Application Profile .....	426

---

<b>8</b>	<b>Automating Processes .....</b>	<b>428</b>
8.1	Creating and Using Macros .....	428
	Using Macros to Automate Tasks .....	428
	Protection from Documents That May Contain Viruses .....	429
	Digital Signatures .....	430
	Adding a Macro Developer to the List of Trusted Publishers .....	431
	Changing the Security Level for Macro Virus Protection .....	432
	Creating a Macro .....	432
	Tips for Recording a Macro .....	433
	Editing a Macro .....	433
	Running a Macro .....	434
	Deleting a Macro .....	434
	Copying a Macro Project .....	435
	Renaming a Macro Project .....	435
	Deleting a Macro Project .....	435
	Digitally Signing a Macro Project .....	436
	Obtaining a Digital Certificate .....	436
	Removing a Macro Developer from the List of Trusted Publishers .....	438
	Troubleshooting Macro Warnings and Security Levels .....	438
	Troubleshooting When Recording and Running Macros .....	441
	Security Levels in FlexPro .....	442
8.2	Automation Using FlexPro Visual Basic .....	443
	Getting Started with FlexPro Visual Basic .....	443
	Working with Objects .....	455
	Working with Events .....	465
	Working with Macros .....	469
	Working with Command Bars .....	473
	Examples .....	481
8.3	Additional Options .....	498
	Remote Control of FlexPro via FPAccess .....	498
	Remote Control of FlexPro via DDE .....	504
<b>9</b>	<b>End User License Agreement .....</b>	<b>507</b>
9.1	FlexPro End User License Agreement (DE AT CH) .....	507
9.2	FlexPro End User License Agreement (International) .....	513
	<b>Index .....</b>	<b>519</b>



# 1 Introduction

## 1.1 Welcome to FlexPro 10

You will soon discover that by acquiring FlexPro, you have found a software tool that is not only extremely efficient, but it is also easy to use due to its versatility. Thus, you will discover many functions by simply using your intuition as you explore FlexPro. FlexPro offers you a logical structure and user interface that is easy to use, in addition to consistent implementation of standard Windows operations, which you are certain to recognize from other programs such as the Microsoft Office family of products.

You can also find all of the information and instructions in this manual in the extensive FlexPro Online Help, which you can access at any time by pressing the F1 key. This printed manual should provide you with an introduction to working with FlexPro regardless of the computer environment.

If you would like to spend time thoroughly learning about FlexPro, we suggest that you read all of the chapters of this manual in order. You can skip the **Working with...** sections the first time you read the documentation. You can find additional tutorials on how to use FlexPro in the Online Help. The most important section that we recommend for those new to FlexPro is **Getting to Know FlexPro in just 15 Minutes**.

We wish you success in working with FlexPro!

The Team at Weisang

## 1.2 Comparison of FlexPro Editions

### FlexPro Developer Suite

FlexPro Developer Suite covers data organization, presentation, analysis and automation and is like FlexPro Professional with all options included.

### FlexPro Professional

FlexPro Professional covers presentation, analysis and automation. Only FlexPro Professional can calculate analyses, preview images and data statistics in the background and parallelizes vector arithmetical operations and object updating on multicore systems. This edition of FlexPro allows you to analyze videos recorded in conjunction with measurement data. FlexPro Professional is the best edition for working in a team, since this is the only edition that supports shared template databases that you can use to access analysis templates and algorithms deployed throughout your enterprise. It is the

only FlexPro edition that contains the Microsoft Visual Basic for Applications (VBA) development system, not only allowing you to record and run macros, but also to write your own programs and dialog boxes. FlexPro Professional also provides access to the FlexPro Automation object model, which you can use to control all editions of FlexPro remotely. It also contains a toolkit that you can use to program your own import filters and FPScript functions. Programs written with FlexPro Professional can also be used for other editions of FlexPro.

### **FlexPro Standard**

FlexPro Standard covers presentation and analysis. Its basic set of features includes a wide variety of analysis procedures. Other analysis procedures are available as optional add-ons. You can also record and run macros. The VBA development environment, however, is not part of this edition.

### **FlexPro View**

FlexPro View offers all of the presentation images included in FlexPro Standard. The analysis functions, the user administration and the customizable user interface are restricted. The following list gives you an overview:

- FlexPro View only supplies the following analysis objects: Statistical Quantity, Minimum, Maximum, Mean and RMS.
- The formula functions are limited to the following: Signal, Derivative, Integral, Mean, Variance, Minimum, Maximum, Range, Reduce, Sum, NumberOfColumns, NumberOfRows, NumberOfElements, ListElementName, HasVoidValues, DataOrder, Increment, Day, Hour, Minute, Month, Second, Weekday, Year, DayOfYear, CurrentDate, Date, Time, Duration, PrimaryListElement, Bit, XScale, Reshape, ValuesInInterval.
- In formulas, the operators for comparisons, Boolean operators and the indirection operator are not available.
- There are no properties available to access objects in formulas.
- The user profile cannot be exported, and user rights cannot be edited.
- The user interface cannot be customized.
- No custom coordinates are available in the coordinates window when working with cursors.
- The function Dimensioning curves is not available.
- Using FlexPro View, project databases created with FlexPro Professional or FlexPro Standard can only be opened to a limited extent.

## FlexPro Reader

You can use the free FlexPro Reader to open project databases created with one of the FlexPro editions mentioned above as read only and view the results. You can display diagrams, documents and other objects and use cursors to view the data inside. The automation interface is not available in FlexPro Reader, and a watermark is placed on expressions.

## Options

You can expand the range of FlexPro Standard and Professional mathematical analysis procedures using the following **options**:

Acoustics Option

Digital Filters Option

Human Body Vibrations Option

Count Option

Order Tracking Option

Spectral Analysis Option

Statistics Option

ASAM ODS Data Import Option

Data Explorer Option

## 1.3 Installation

FlexPro was developed and tested for Windows 2003 Server, Windows 2008 Server, Windows Vista, Windows 7 and Windows 8 operating systems. Before installation, you should make sure that your operating system has the latest service pack from Microsoft installed. By selecting the **System** icon in Windows Control Panel, you can check to see which service pack is installed on your computer. Microsoft offers the service packs as a free download on the Windows Update web site. To display Online Help, FlexPro requires Microsoft Internet Explorer version 4.02 or higher. To install FlexPro, use the installation program **Setup.exe**.

## System Requirements

To run FlexPro 10, the system must meet the following minimum requirements:  
Windows 2008 Server, Windows 2012 Server, Windows Vista, Windows 7 or Windows 8

operating system, Pentium processor (Dual Core or higher recommended), 512 MB (1024 MB recommended) or 1024 MB (2048 MB recommended) RAM for Windows Vista, Windows 7 or Windows 8, and 300 MB of available hard disk space (1 GB recommended).

For users working with very large amounts of data, such as for spectral analysis of acceleration signals, we recommend using FlexPro Professional on a quad-core or higher multicore system with 8 GB of RAM, the Windows 7 64-bit operating system and a 15,000 RPM hard drive or SSD.

### FlexPro 10 Silent Setup

#### Preparation

The following steps are required in order to prepare for a silent/unattended setup of FlexPro 10:

- Run an administrative setup of FlexPro:

```
setup.exe -a
```

This is not the actual installation, since the setup files are only prepared for installation and uncompressed to a folder, preferably a network share. The setup wizard prompts for the destination directory.

- Next, you need to carry out a normal installation from there. The settings selected in the wizard are stored in a response file (.ISS extension). To ensure that response file will be recorded, the setup must be launched using the following parameters:

```
<Network Share>\Setup.exe -r -f1<Network Share>\setup.iss>
```

The parameter -r stands for recording, -f1<...> specifies the name and path of the response file. If -f1<...> is not specified, the setup.iss response file is placed in the Windows folder (typically C:\Windows).

---

**Note:** Silent installation is only possible using setup routine on the CD provided by FlexPro. The packed, downloadable setup cannot be used for this type of installation.

---

#### Client Installation

After the setup files and response file were made accessible to all clients on a network share, the client installation can be launched manually or via relevant deployment tools as follows:

```
<Network Share>\Setup.exe -s -f1<Network Share>\setup.iss [-f2<path to Log folder\FlexPro10-<ClientName>-Setup.log>]
```

The parameter -s stands for silent. FlexPro is then installed on the client based on the

settings in the response file without displaying a user interface.

If a user uses the optional parameter `-f2<...>`, a log file is created, which records the installation process.

### **Licenses**

During setup, no license information is requested, nor is it installed.

Activation of the single user license currently takes place on the computer when the application is first launched.

No additional action is required for network licenses if the license server RLM.exe is on the same network segment as the clients. In this case, the license server is found automatically.

Otherwise, the connection data can be specified with the following registry entry:

HKEY\_CURRENT\_USER\Software\Weisang\FlexPro\10\License\RLMLicenseServerName  
(REG\_SZ) = Host name or IP address of computer on which RLM.exe is running.

HKEY\_CURRENT\_USER\Software\Weisang\FlexPro\10\License\RLMLicenseServerPort  
(REG\_DWORD) = Port, usually 5053.

### **References**

A detailed description of the InstallShield setup.exe parameters are available under:

Q105473: INFO: Setup.exe Command-Line Parameters

<http://kb.flexerasoftware.com/selfservice/microsites/search.do?cmd=displayKC&docType=kc&externalId=Q105473>

## **1.4 License Manager**

The FlexPro License Manager shows you what FlexPro licenses are available on your computer and network and allows you to select a license and to specify the preferred FlexPro configuration, consisting of the edition and options.

### **Single User License**

When launching FlexPro after completing the installation, an evaluation license is installed on your computer, which gives you 30 days to try it out. An Internet connection is required in order to download the evaluation license. You will need to activate the license as soon as the evaluation period expires in order to continue using the complete version of FlexPro. Without a valid license, FlexPro runs in reader mode. Activation will permanently link the purchased single user license to the computer. To activate the license, use the product key supplied with FlexPro. This will allow you to activate the software for personal use up to two times. For instance, you can install and activate

FlexPro on your desktop computer and on a laptop. You can activate FlexPro online by communicating directly with the Weisang activation server, or you can activate it offline by swapping files. You can also deactivate a license. The license will then be credited to your product key and you can install FlexPro on a different computer.

---

**Important note:** Single user licenses cannot be used on a virtual machine or a terminal server session. If you install a single user license on a virtual machine, it will be activated there, but will not work. You will lose one activation.

---

### Network License

A network license, also known as a concurrent use license or floating license, is usually installed on a server by the network administrator and limits the number of concurrent users. When launching FlexPro, the license is requested by the server, and when you close FlexPro, the license is then available once more to other users. If you have set up FlexPro to use a network license, you must connect to this server as long as FlexPro is running. There are also FlexPro network licenses available that let you check out, i.e. transfer, the license to your local computer for a period of 120 days.

### Dongle-based Licenses

Weisang can also provide the license with a USB dongle for an extra charge. This type of license is not tied to the computer, but is instead tied to the USB dongle and can be moved to a different computer by simply inserting the dongle into a USB port.

---

**Note:** More information about the License Manager is available in the document **Installing and Using the FlexPro License Manager** which is located in the **License Manager** folder on the FlexPro CD.

---

## 1.5 Working with the License Manager

### Opening the License Manager

- Select **License Manager** from the ? FlexPro menu.

---

**Note:** During the evaluation period and after a license has expired, the License Manager will appear automatically when FlexPro is started.

---

## Online Activation

1. Launch FlexPro and open the License Manager. To do this, in the ? FlexPro menu click on **License Manager**.
2. Click **Local**.

If you have a dongle-based license:

3. Insert the USB dongle into an available USB port on your computer.
4. Wait until the USB dongle driver finishes installing and the red LED on the dongle lights up.
5. In the **FlexPro License Manager** dialog box, click **Extended** and check the first entry in the **Show Host IDs** dialog box when it appears. The entry must be **HASPHL=xxxxxxxxxx**.
6. When online activation is selected, a connection is established with the **rlm.igb.weisang.com** Weisang activation server. If you want to connect using a proxy server, click **Proxy Server** to enter the login information.
7. Under **Product key** enter the product key for FlexPro and for all the options that you want to install.
8. Now click on **Activate**.
9. The dialog that appears next will show you all the licenses that will be installed once you complete activation. Check the list carefully and click **Cancel** if you see any errors. Otherwise, click **OK** to continue with activation.
10. After activation has been confirmed, your available license will be displayed. Select the edition and options that you want to use and close the License Manager by clicking **OK**.

If online activation fails, FlexPro will display an error message. The cause is usually a block in communication between the Weisang activation server and your computer. If this is the case, try Offline Activation.

## Offline Activation

To activate a license offline, i.e. without connecting to the Internet, you will need to provide Weisang with what is called the host ID of your computer or your dongle in addition to the product key for the license. The host ID identifies your computer/dongle and the license is tied to the computer/dongle by way of the host ID.

If you have a dongle-based license:

1. Insert the USB dongle into an available USB port on your computer.
2. Wait until the USB dongle driver finishes installing and the red LED on the dongle lights up.
3. Launch FlexPro and open the License Manager. To do this, in the ?FlexPro menu click on **License Manager**.
4. In the License Manager, click **Extended**.
5. In the **Show Host IDs** dialog box that appears you will see the host IDs. If you have connected the dongle and it is detected correctly, the first entry should be **HASPHL=xxxxxxxxxx**. Click **Copy to the Windows clipboard**.
6. Now send an e-mail to **licensing@weisang.com**. Include in the e-mail the displayed host IDs (paste from the Clipboard using CTRL+V) and the product keys that were provided to you with FlexPro. Please note that your e-mail must include a valid e-mail address for the reply. After a Weisang representative has activated your license, you will receive an e-mail with a license file (.lic) attached.
7. Open the folder C:\ProgramData\Weisang\FlexPro and copy the license file into this folder.  
**Notes:** The ProgramData folder is not usually displayed in Windows Explorer. You should therefore enter the entire path name using the keyboard. If a file with the same name is already in the folder, rename the new file before copying it to the folder.

After you have copied the file, open the License Manager from the ? FlexPro menu. FlexPro will then import the license and will display the available licenses. Select the edition and options that you want to use and close the License Manager by clicking **OK**.

### Checking Out a Network License

1. Open the License Manager and
2. Make sure that the **Network** tab is active and that the specified edition and selected options correspond with the configuration that you want to use offline.
3. If the network license can be checked out, the **Check Out** button appears. Click on this button.
4. In the dialog box that appears, enter the number of days that you wish to check out the license. This can be any length of time up to 120 days. After this period has expired, the local license will be deactivated automatically and the license will be



available on the server again.

5. Click **OK** to check out the license.
6. If a confirmation message appears, the FlexPro License Manager will automatically switch the license to **Local** and then you can disconnect the computer from the network. Otherwise, you will have to repeat the process by selecting a shorter checkout period.

## **Returning a Checked Out License Early**

1. Make sure that your computer is connected to the license server network.
2. Open the License Manager and make sure that the **Local** tab is active.
3. If the local license has been checked out, the **Cancel** button appears. Click on this button.
4. FlexPro then returns the license to the server and the License Manager converts it back to a network license. To close the confirmation dialog box and the License Manager, click **OK**.

## **Transferring a License**

### **a) License without a USB dongle**

Before transferring a local software license from one computer to another, deactivate it on the original computer and then reactivate it on the new system. This procedure requires that you have an active maintenance agreement with Weisang for the license.

1. Open the License Manager and make sure that the **Local** tab is active.
2. In the License Manager, click **Deactivate**.
3. In the dialog box that appears, click **OK** to confirm that you want to deactivate the license.
4. You can now install the license(s) on another computer by reusing the product key.

### **b) License with a USB dongle**

This type of license is not tied to the computer, but is instead tied to the USB dongle. This means you do not need to deactivate the license.

1. Insert the USB dongle into an available USB port on the computer to which you would like to transfer the license. As soon as you have an Internet connection, Windows will

automatically download and install the driver for the USB dongle. Your other option is to install the driver located on the FlexPro CD. More information about this is available in the document **Installing and Using the FlexPro License Manager** which is located in the **LicenseManager** folder on the FlexPro CD. If Windows can read the dongle correctly, a red LED lights up on the dongle.

2. Now all you have to do is transfer the license that is tied to the dongle to your computer. You can do this online by entering the product key that came with the dongle in the FlexPro License Manager on the new computer and then clicking **Activate**. Alternatively, you can copy the license file from C:\ProgramData\Weisang\FlexPro\FlexPro.lic to the same folder on the new computer.

## Troubleshooting

A failed installation or License Manager setup usually manifests itself when FlexPro cannot find a license when launched. When an error occurs, a small button with a warning triangle appears in the License Manager. If the button appears, click on it to find out more about the error.

### Check if the license file is installed correctly on the local computer:

The folder C:\ProgramData\Weisang\FlexPro must contain at least one file with the .lic extension. The file should contain at least one license entry that looks similar to the following:

```
LICENSE weisang flexprodevelopersuite 10.0 permanent uncounted
  hostid=rehost=111e02d861211656b958ed82e2b254fc6ebc8608
  _ck=6708fccc64 sig="60P0450WF36CP91AQQNFHB82YFN7ENN6J1XP8WG22H6PTXS
GM7243KM4X44HP863EKTDNJA4R"
```

### Checking the Host ID

FlexPro single user licenses use a "re-hostable" host ID. This host ID is created when a license is activated and is deleted when the license is deactivated.

The folder C:\ProgramData\Reprise\weisang must contain at least one folder called "License". In the example above, this would be flexprodevelopersuite.

To prevent bypassing the activation mechanism, the re-hostable host ID is read only. It is not possible to restore it using a backup program.

If no host ID is available, you will have to carry out activation again. If the host ID is available, but you are unsure of its integrity, please contact Weisang customer support

or the retailer where you purchased FlexPro.

### **Additional Support**

Information on troubleshooting FlexPro network licenses is available in the document **Installing and Using the FlexPro License Manager** which is located in the **License Manager** folder on the FlexPro CD.

If you need additional help with troubleshooting, send an e-mail with your contact information as well as your FlexPro serial number and product key to **support@weisang.com**. The serial number and product key are located in the FlexPro CD case. We will then contact you to help you solve the problem.

## **2 Tutorials**

### **2.1 For Newcomers**

#### **Getting to Know FlexPro in just 15 Minutes**

This tutorial provides you with a brief overview of the structure of FlexPro and the basic command methods.

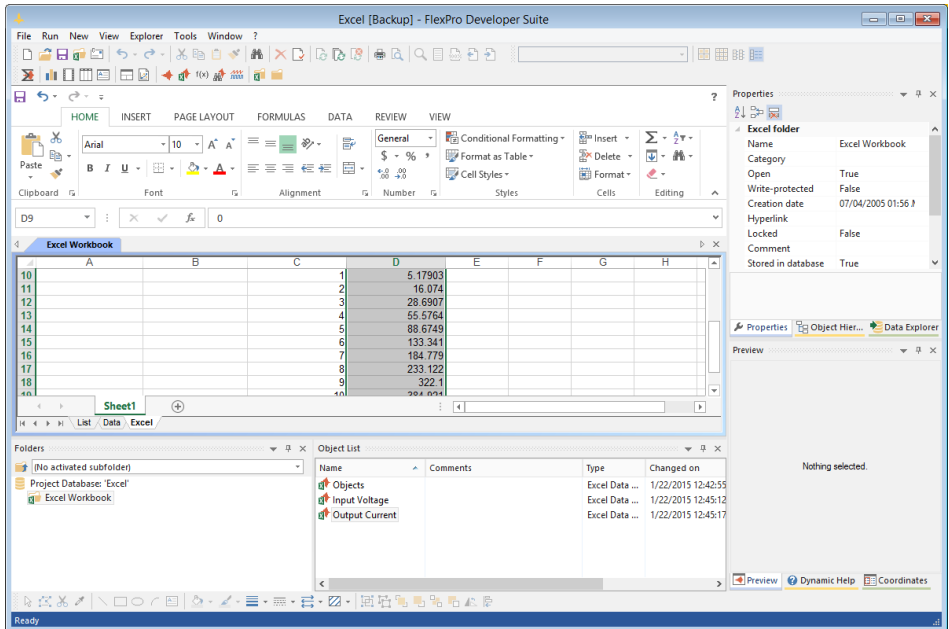
You can find this information in the FlexPro Online Help and as a separate booklet included with FlexPro.

#### **Tips for New Users**

If you already use a spreadsheet program such as Microsoft Excel, then the information in the following sections will be very useful. Here we show you how you can more easily and efficiently manage operations familiar to you using FlexPro.

#### **Data Management**

When working with a spreadsheet, you usually arrange all of the data that you want to analyze in a table. Scalar values, such as the serial number or the date of a measurement, occupy one cell in the table. The results of a series of measurements are usually arranged in one or more columns.

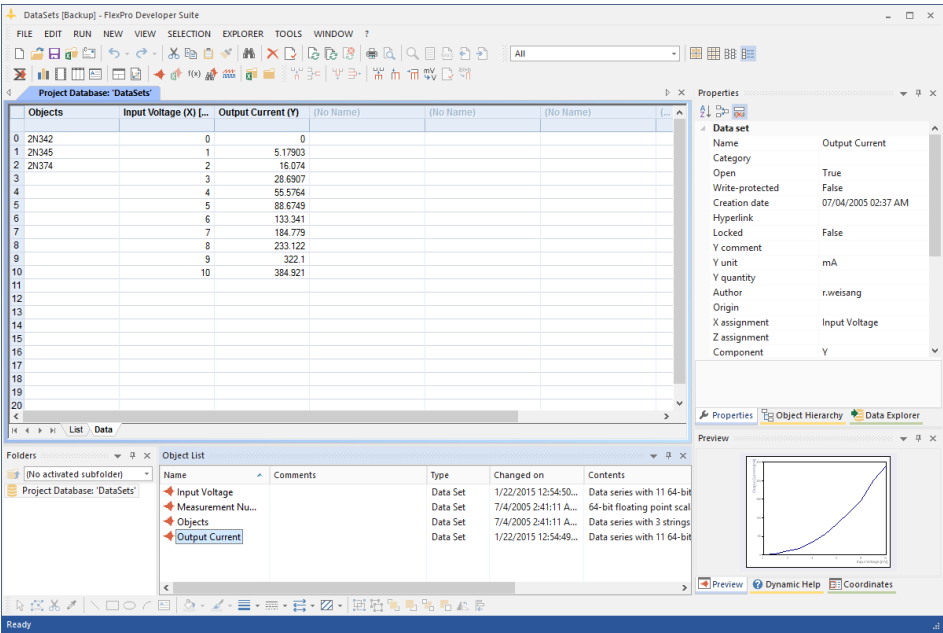


As long as the data volumes are small, this is very practical, since you have direct access to all of the data. However, this procedure is limited when you are working with large volumes of data or when the amount of data varies, as in a measurement series, for example.

In developing FlexPro, however, situations like these were taken into account. Data in FlexPro is therefore divided into multiple data sets that are stored together in a FlexPro project database. In this case, each data set only accepts data of the same type. If you look at the example table more closely, it is clear that it contains data with different data structures and data types. For example, the measurement number is a scalar value, the list with the samples is a data series with strings, and the input voltage and the output current are both data series, each with eleven values. As you can see, the data structure specifies how associated data is organized, and the data type specifies whether the data consists of floating point values, strings or other values.

For each element in the example, FlexPro provides a matching data structure used for the corresponding data set. For example, the serial number of the measurement is saved as a data set **Measurement Number** with a scalar value. The list contains the test objects as an objects data set with a data series containing several strings and the **Input Voltage** and the **Output Current** as data series. In the illustration above you see an Excel workbook opened in FlexPro and the assigned data sets. In this example, the data

sets are Excel data links, which means that they do not contain the data directly, but refer to the corresponding areas in the Excel workbook. You can convert these types of links into FlexPro data sets at any time. You then can edit data set values using FlexPro.



As the above picture illustrates, FlexPro can display data series similarly to Excel in a data grid, and you can easily edit them.

To work seamlessly with FlexPro, it is very important that you recognize which data structures are present. Often one is not aware of this when working with a spreadsheet, since the data structures implicitly result from the way the data is arranged in the table cells. A common mistake, for instance, is to place adjacent columns in a single data set, even though the columns contain different values. The following table will help you determine the correct data structure of the various elements in your spreadsheet.

FlexPro Data Structure	Description	Appearance in Spreadsheet
Scalar value	A scalar value, such as the date of a measurement, a single measurement value or another parameter	A cell containing a value, such as the measurement number in the above example.
Data series	A column with several associated values. All values have to have the	A column with several cells, such as the input voltages and output

	same physical unit.	currents in the above example
Data matrix	A matrix of associated values. All values have to have the same physical unit.	Several columns adjacent to each other. These matrices are used for surface graphics, for instance.

The data itself often also contains additional information, such as the physical unit or comments. This is saved as header information in the data sets. When the data is later displayed, FlexPro can access it and automatically label the diagram axes, for example.

Data Set 'Characteristic Curve' Properties

General  
Parameters  
Component & Assignments  
Formatting  
Data Types

Comments

Y: 
X:

Unit

Y: 
X:

Quantity

Y: 
X:

Lower range limit

Upper range limit

Timestamp

Y: 
X:

Origin

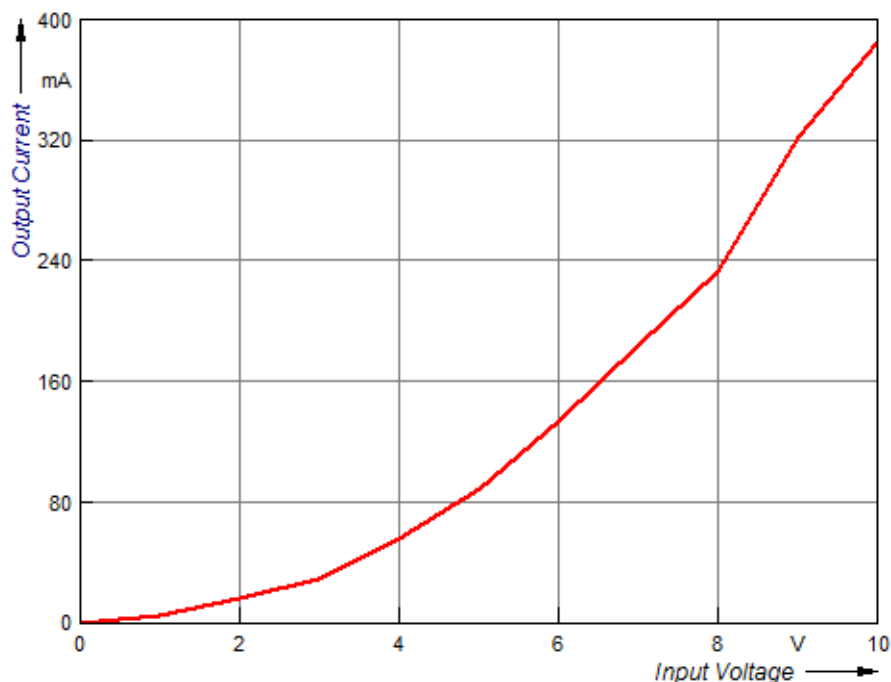
Hyperlink

Created on 7/4/2005 2:43:58 AM by  
Signal with 11 64-bit floating point points

## Signals

If you display the output current as a curve in a diagram in your spreadsheet, then the indices of the measurement values are first displayed on the X axis of the diagram. However, it makes more sense to display the measured characteristic curve, which means that you would have to write the input voltages belonging to the output currents measured onto the X axis. The two data series **Input Voltage** and **Input Current** thus together form a signal.

23



Signals very often occur in measurement applications, for instance, whenever you measure a time series. The times at which the measurement values were sampled form the X component, and the measurement values themselves form the Y component of the signal. Since signals occur so frequently, FlexPro offers you signals in their own data structure. A data set containing a signal consists of two data series of equal length, which are called X component and Y component.

If you display a data series in a diagram as a curve, then the X values are numbered consecutively, as is also the case in the spreadsheet. However, if you display a signal as a curve, the X values belonging to the signal's Y values are automatically displayed on the X axis. Analyzing signals is also easier than analyzing separate measurement series. An integration is then written as `Integral('Characteristic Curve')` instead of `Integral('Output Current', 'Input Voltage')`. You can access one of the signal components any time: `Signal.x` provides the X values and `Signal.y` provides the Y values. You can specify individual comments and the individual physical unit for each component in the data set. FlexPro recognizes more data structures than just the "Signal" data structure presented here. You can find out more under Data Structures.



FlexPro usually creates signals as soon as data are imported. If, for instance, you import two columns from an Excel workbook, FlexPro asks whether you want to import the two columns as a signal or as two separate data series.

Instead of creating signals, you can also assign data series to each other. Simply enter the data set with the X values as an assigned data set into the data set with the Y values. FlexPro can then automatically adopt the X data set when a diagram is created, for instance.

## Editing Data

When you open a FlexPro data set, its contents are displayed as a table in the data set window, allowing you to carry out your editing in the manner with which you are already familiar. FlexPro can display all of a folder's data sets with data series and data matrix data structures next to one another in the folder's Data View. It is not necessary for you to open them in separate windows.

The following picture shows a signal opened in the data set editor:

	X	Y
0	0	0
1	1	5.17903
2	2	16.074
3	3	28.6907
4	4	55.5764
5	5	88.6749
6	6	133.341
7	7	184.779
8	8	233.122
9	9	322
10	10	384.921
11		

## Importing Files

To import data into a spreadsheet, you can read the data from a text file, for instance, which will then be more or less copied "as is" into the spreadsheet. FlexPro has an Import Wizard for various file formats that can provide you with step-by-step assistance with importing.

If you work with special file formats, such as a recorder's binary files, the wizard can offer you the complete contents directly for you to choose from and you can select the

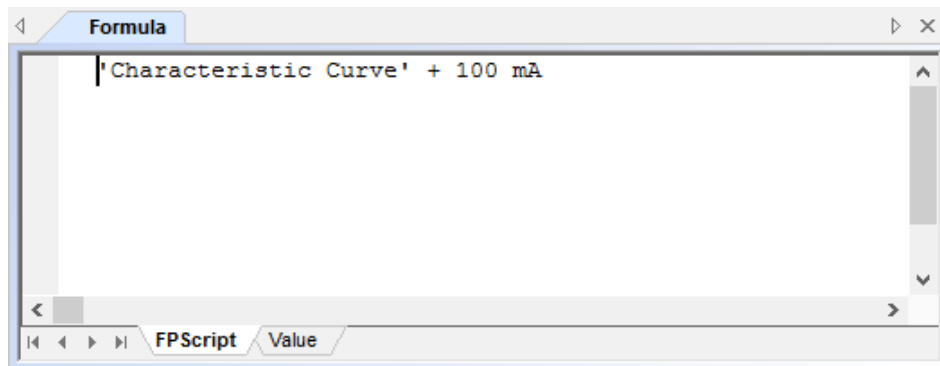
data sets to be imported from a list. FlexPro also transfers information such as units or comments. With unspecified formats, such as, for instance, data in text format, you have to provide the Wizard with some information on the file structure, which you may already be familiar with from working with spreadsheets. The result, however, is always a series of data sets that are displayed in the Object List.

The Import Wizard can copy the data into the FlexPro project database or create links to the original data. If you work with links, changes to the original files immediately affect the analysis results in FlexPro. FlexPro can cooperate particularly closely with Microsoft Excel. You can import entire Excel workbooks into the FlexPro project database and work with Excel in the project database. To access data in the Excel workbook, create Excel data links.

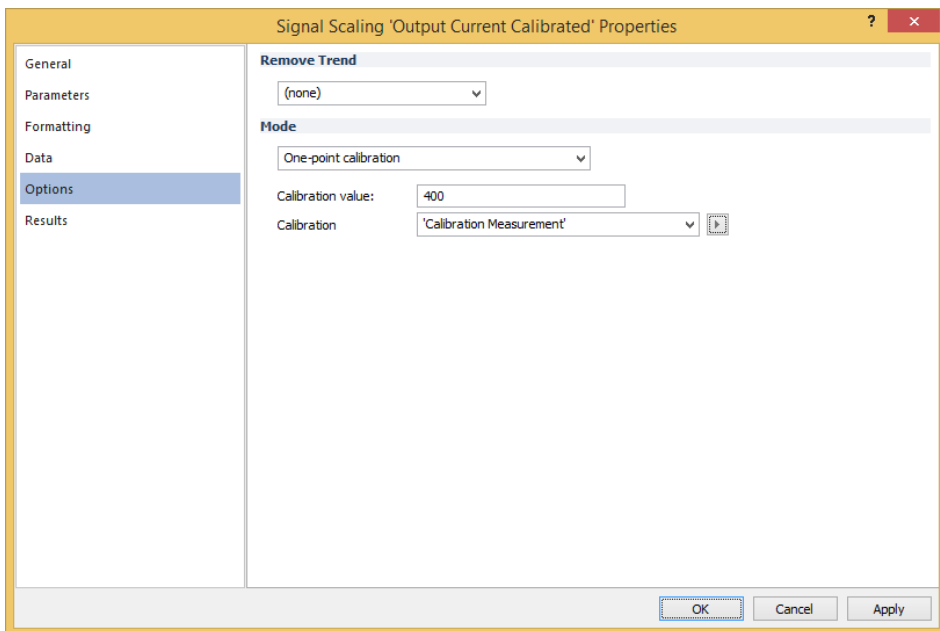
FlexPro can exchange data directly with other programs as well. You can import and export data from other programs. You can exchange data dynamically in both directions using FlexPro's OLE data link.

### Using Data in Calculations

If you use data in spreadsheet calculations, then you enter a formula in the cell in which the result is to appear. This formula refers to the cells to be used in the calculation. The process is similar when working with FlexPro. In the project database, you set up formulas that perform the desired calculations. In addition to providing scalar values as the result for formulas that you set up, FlexPro can also return data series, signals, etc. When the formula is calculated, FlexPro automatically determines the data structure of the result, which depends upon the elements to be used in the calculation. For example, the formula can be a simple arithmetical expression. The formula 'Characteristic Curve' + 100 mA adds, for instance, an offset of 100 mA to the current values contained in the Y component. The result is once again a signal, and you do not have to worry about the X component, which is very practical.



However, you can also write proper programs in formulas. To do this, use the FScript language. You do not even have to write a formula for common analyses. FlexPro offers special analysis objects for this purpose. Technically, these are also formulas that FlexPro writes for you. You simply select the analysis options in a dialog box. And if you change these later, the formula is corrected automatically.



When using formulas, it is important to know that you can use formulas like data sets. For example, you can calculate the results of other formulas within a formula or display a formula as a curve in a diagram. Formulas even have the same header information as data sets.

### Selecting Data Ranges

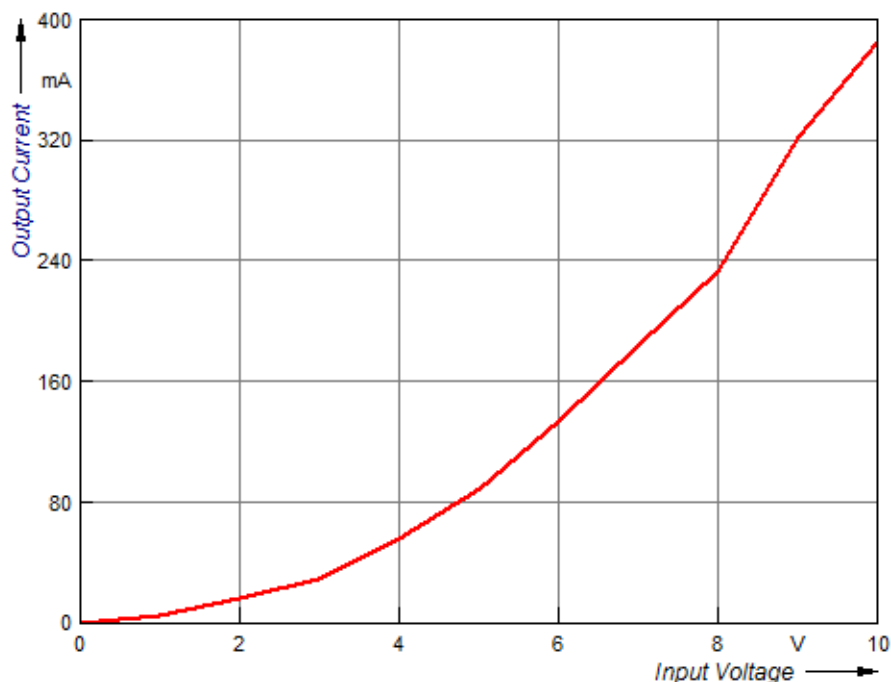
You can access sections of data in the spreadsheet by referring to cell ranges. FlexPro offers an index operation for this with which you can extract individual values or ranges of data from a data set or from the result of a formula. The formula 'Characteristic Curve' [0, 9], for example, extracts a partial signal with the first 10 points from the characteristic curve.

These index operations are automatically created by FlexPro if, for instance, during the creation of a diagram you do not select entire data sets, but only a cell range in a data

set window. The cursors offer a **Copy Range** command with which you can create interactive ranges from data sets.

### Displaying Data Graphically

You can place diagrams onto a sheet to display data in a spreadsheet graphically. You therefore specify the cell ranges for the Y values and possibly also the X values for the curves in the diagram. The process is similar when working with FlexPro. You set up a diagram in the project database using the Diagram Wizard and in the process specify the data sets to be displayed as curves in the diagram. Later you can add more curves by simply dragging and dropping data sets or formulas into the diagram. When doing this, FlexPro scales and labels the axes automatically, and comments, units, etc. are taken from the data set header information.



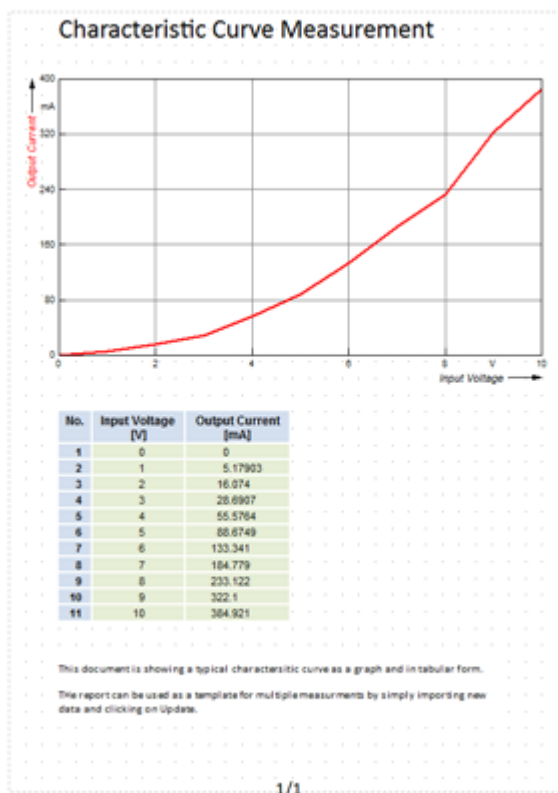
In addition to 2D and 3D diagrams, FlexPro offers you additional formatted column tables, cell tables and texts into which you can embed scalar values to display your results.

No.	Input Voltage [V]	Output Current [mA]
1	0	0
2	1	5.17903
3	2	16.074
4	3	28.6907
5	4	55.5764
6	5	88.6749
7	6	133.341
8	7	184.779
9	8	233.122
10	9	322.1
11	10	384.921

### Documentation

To document the results, you usually set up a table in a spreadsheet where only the relevant data, diagrams, etc. are displayed. To do this, FlexPro has a powerful Document Editor that you can use to create illustrative, multi-page documents.

In addition to your own drawings and texts, you can insert links to diagrams, tables and texts that you have set up in the FlexPro project database into these documents. Alternatively, you can also embed these objects directly into the document without having them appear in the project database. Depending on your needs, FlexPro can automatically wrap diagrams and tables across several document pages. You create the document in the project database just like you would other objects.



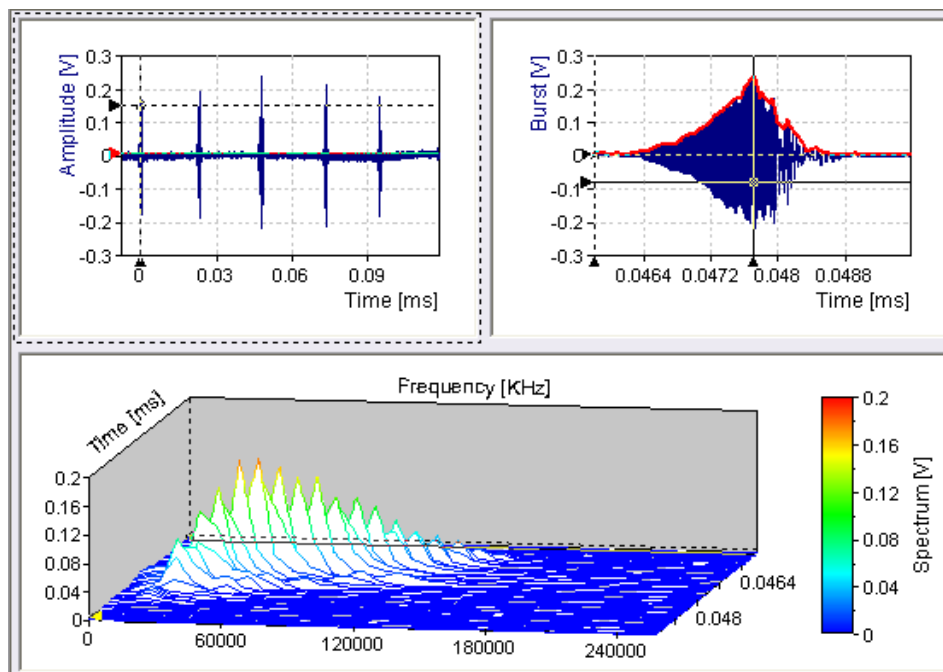
## Importing and Exporting Graphics

If you write large reports, you may want to insert a diagram or table into a Word document, for example. This is no problem with FlexPro. You can create the diagrams in the project database and use your mouse to drag a copy into your Word document -- done! You can also insert graphics from other programs into the documents created using FlexPro.

## Visually Analyzing Data

If you would like to interactively analyze data in a spreadsheet, you often have no choice but to look at the columns of values or the diagrams created. With FlexPro you can measure the curves in your diagram using cursors. Cursors work in 2D and 3D diagrams. You can zoom into sections, edit data, cut out areas and much more. The values under the cursors are displayed in a coordinates window. As an advanced FlexPro user, you can tailor the contents of the coordinates window to fit your needs.

FlexPro also allows you to synchronize cursors across several diagrams. To do this, use the worksheet. This is a window that can be split into up to 16 panes. You can insert diagrams into these panes and measure the curves of the diagrams with the cursors.



In addition, anything you can do in a worksheet can also be done in a document. However, the worksheet is often more practical for working onscreen.

## **Analysis Automation**

To automate an evaluation with a spreadsheet, you usually have to write a program that controls the necessary steps.

FlexPro offers automation without any programming. Due to the fact that all calculation processes are permanently saved in formulas and analysis objects, you can easily apply them to several measurements, for example. Set up folders in the FlexPro project database where you want to save the data from a certain measurement. Before you update a document, for example, just activate the folder from which the data is to be taken. FlexPro then accesses the data from the activated folder and you obtain an evaluation of the selected data at the touch of a button.

In addition, FlexPro has a macro recorder that you can use to record operations as Visual Basic macros. With the automation interface, you can integrate FlexPro into applications and manage it remotely.

## **What's Next?**

If you have not worked with FlexPro yet, we suggest that you look at the tutorial Getting to Know FlexPro in just 15 Minutes.

## **2.2 Data Analysis**

The data analysis tutorials cover video analysis as well as different analysis procedures supported by FlexPro and various analysis options.

You can find them in FlexPro Online Help.

## **Test Series Analysis Tutorial**

This tutorial shows you how to access the data and attributes of a test series for analysis purposes. You should already have some experience with FlexPro before you work through this tutorial.

Through this tutorial you will learn how to:

- access test series data and attributes using DataQuery,
- carry out analyses for a variable number of data sets,
- display calculation results and attributes in table form,
- exclude outliers from the analysis,



- sort your results by any attribute,
- and use data set parameters in an axis label.

If you have a license for the **Data Explorer option**, you will learn how to access data on your hard drive directly without having to import it first.

---

**Note:** The tutorial result is a project database called `SeriesAnalysis.fpd`, which is included with FlexPro. If you do not want to follow all the steps in the tutorial, you can leave it open and take a look at the objects. The objects are in the folder C:\Users\Public\Documents\Weisang\FlexPro\10\Examples or C:>Users>Public>Public Documents>Weisang>FlexPro>10>Examples.

---

## Analyzing Imported Data

### Setting up indexing

1. Launch FlexPro and choose the **Properties** command from the FlexPro **File** menu.
2. Switch to the tab called **Indexing**.
3. Make sure that **Activate data query indexing** is selected.

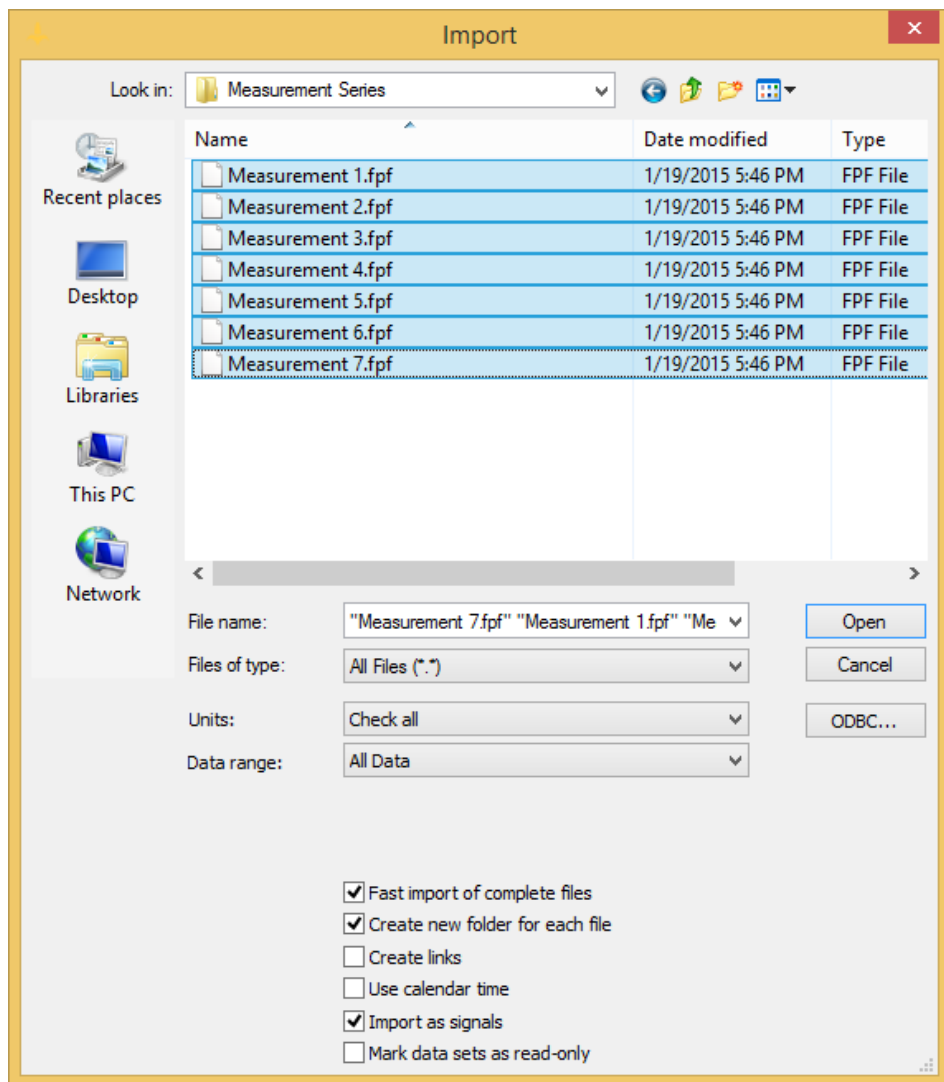
---

**Note:** If the option is not already selected, this may be because the related setting in the **Options** dialog box on the **Project Database** tab under **Default settings for newly created project databases** is not selected. The **Options** dialog box is in the FlexPro **Tools** menu.

---

### Importing the test series data

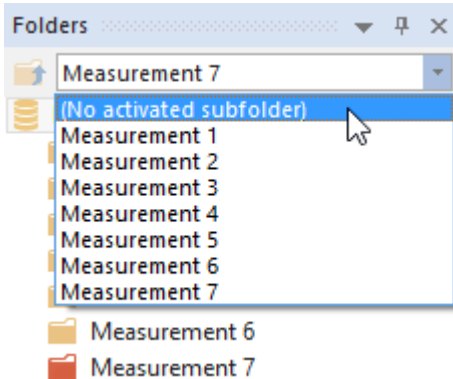
4. Launch FlexPro, right-click with your mouse in the Object List, and select **Import**.
5. The **Import** dialog box will open. Switch to the folder C:\Users\Public\Documents\Weisang\FlexPro\10 or C:>Users>Public>Public Documents>Weisang>FlexPro>10. From there, switch to the subfolder Examples\TestSeries and select the seven files from Measurement 1.fpf through Measurement 7.fpf.
6. Select the options **Create new folder for each file** and **Fast import of complete files**.



7. Click **Open**.

8. The **Verify Units** dialog box which appears shows that the two units G and s are correctly interpreted. Acceleration signals are contained in the files. Click **Close**.

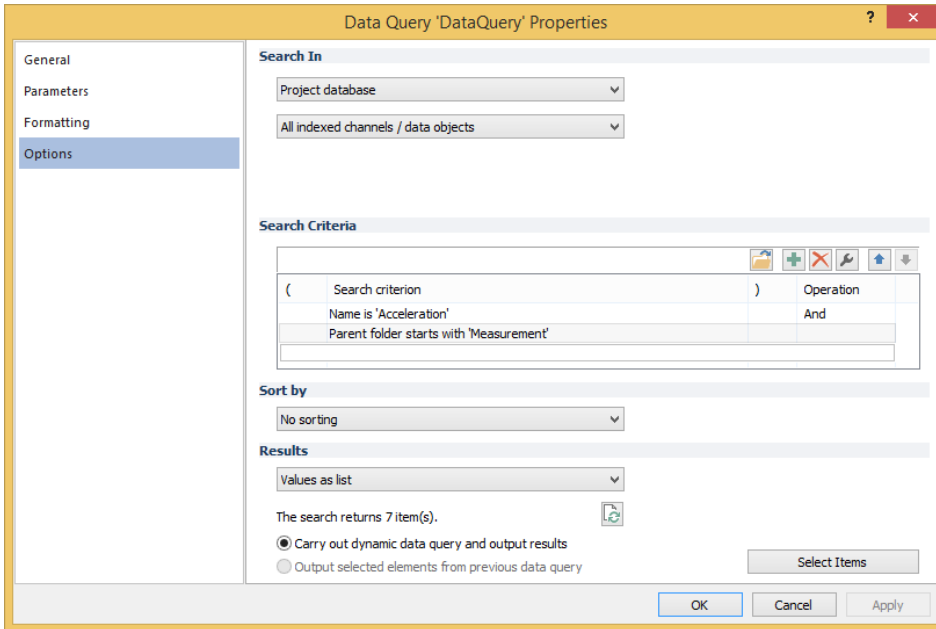
9. The folders **Measurement 1** through **Measurement 7** appear in the Object List. **Measurement 7** is highlighted red because it is activated. To change the focus from this folder, select **(No activated subfolder)** in the list box in the Folders window on the left next to the Object List.



### Carrying out a data query

To access acceleration signals in the project database, use the **DataQuery** object.

1. Right-click with your mouse in the Object List, point to **New** and then click **DataQuery**.
2. The **Data Query Properties** dialog box will open, where you can change the query parameters.
3. Under **Search in** do not change the **Project database** and **All indexed channels / data objects** default settings. This states that the search will take place in the open project database and will search through the entire database.
4. Since you are looking for acceleration signals, the first search criterion should be the name. In the **Search criteria** list, double-click on the existing criterion **Name is**.
5. In the **Edit Search Criterion** dialog box, select **Acceleration** from the list box on the right and click **OK** to close the dialog box.
6. To ensure that only the signals in the measurement folders are found, add one more search criterion. Click on **Add search criterion** and select **Parent folder** for the **Search element**. For the **Search condition**, select **begins with** and for the **Value** enter **Measurement**. Next, click **OK** to close the dialog box.
7. Now you can test your data query by clicking on the **Run data query** button under **Result**. The message "7 item(s) found" should now appear on the left. If not, you will need to check your entries.



8. The option **Values as list** is preselected under Result. This is a list with the data of the acceleration signals. However, you will need not only the data itself, but also other attributes for the analysis. For this reason, the first data query provides only the keys of the found data sets, which will then be used in additional queries to access the data and attributes. You should therefore select **Keys of channels / data objects found**

**Note:** Why channels / data objects? You can use the data query to access data in the project database and external data on your hard drive (the latter requires the **Data Explorer option**). You can access channels on your hard drive that contain files, and you can access data objects (data sets, analyses, formulas) in folders in the project database.

9. Now click **OK** to close the dialog box.

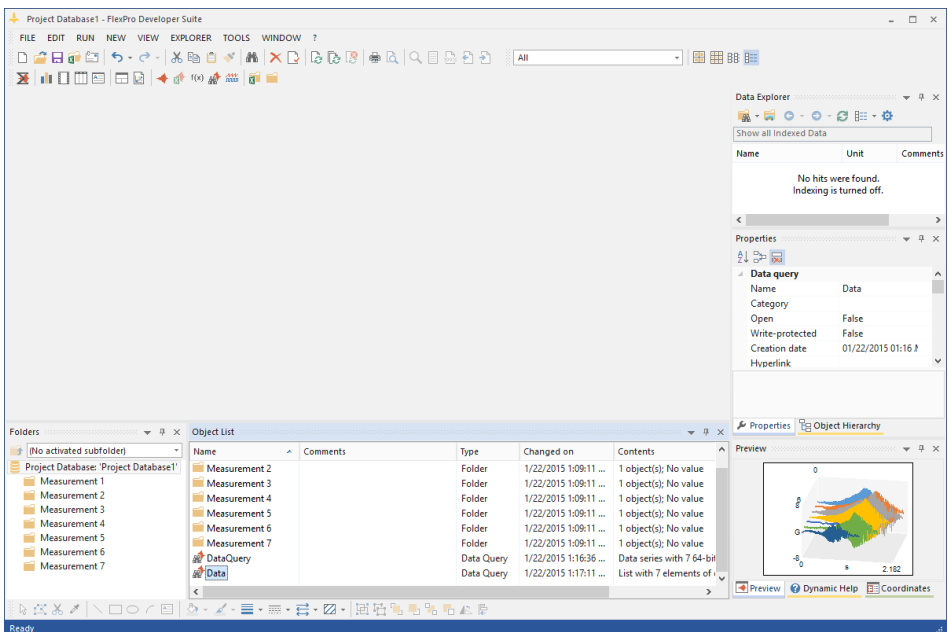
10. Review your data query. The text in the **Contents** column of the Object List is "Data series with 7 64-bit integer values".

### Accessing the data in the found data sets

1. Right-click with your mouse on the data query that you just added to the Object List,

then point to **New** and click **DataQuery** again. FlexPro will now add another data query and will link it to the existing one. In the **Data Query Properties** dialog box, you can see that under **Search** in the option **Channels / data objects with keys** is selected in the second list box, and you will see the previously created data query underneath it. The option **Get result only, no further search** is already selected. This states that no additional search criterion will be used; only a result for the existing data query will be accessed.

2. The **Values as list** option has already been selected for the **result**. Click **OK** to close the dialog box.
3. Rename the newly created **DataQuery2** by pressing F2 and entering "Data". The data will be displayed in the preview as a 3D diagram.

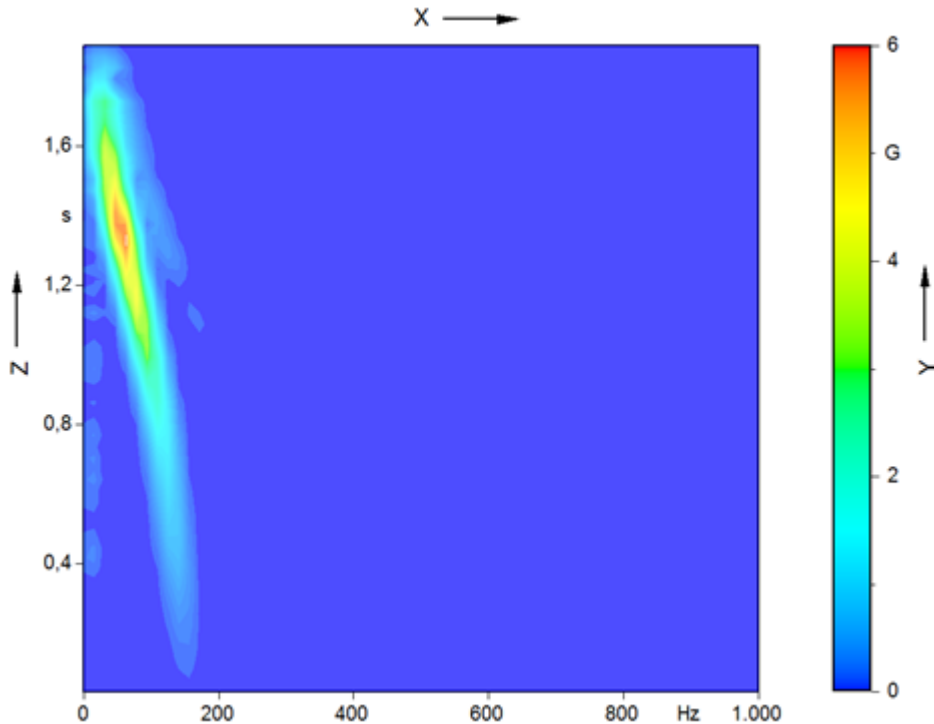


## Analyzing the data

Acceleration signals have a variable frequency and amplitude. In the following, the maximum occurring spectral amplitude for each signal and the frequency at which it occurs will be determined.

The Data object provides the seven acceleration signals as a list with seven items. FlexPro should now run the steps of the analysis as multichannel so that for each step a new list with seven results is generated.

The first step in the analysis is to calculate the time-frequency spectrum for the data. In this case, a two-dimensional data set is generated for each signal, with the frequency in the X component, time in the Z component and amplitude in the Y component, as shown here:



4. Select the **Data** object from the Object List. From the FlexPro **New** menu, select **Spectral Analysis** and from within that select **Time-Frequency Spectrum**.
5. In the **Select List Items** dialog box that appears (because the Data object produces a list with multiple data sets), do not change the **Complete list** setting; just click **OK**.
6. In the **Spectral Analysis** dialog box, select **Amplitude** under **Spectrum Type**. You can leave all other settings unchanged.

Time-Frequency Spectral Analysis 'DataSpectrum' Properties

**General**  
Parameters  
Formatting  
Data  
**Options**  
Results

**Spectral Procedure**

☒ Short-Time Fourier Transform (STFT) ☐ Peak-Hold Short-Time Fourier Transform  
☐ Continuous Wavelet Transform (CWT)

**Spectrum Type**

☒ Amplitude ☐ Normalized dB ☐ SSA - Sum Amplitude<sup>2</sup>  
☐ RMS Amplitude ☐ PSD ☐ Variance  
☐ Amplitude<sup>2</sup> ☐ TISA - Time Integral Amplitude ☐ Magnitude<sup>2</sup>  
☐ dB, Reference: 1 ☐ MSA - Mean Amplitude<sup>2</sup> ☐ Magnitude

**Window**

Type: Kaiser-Bessel Normalization: Amplitude  
Adjustment: 3

**Parameters**

FFT length: segment length Overlap %: 50  
Segment length: sqrt(data length) \* 2

**Options**

Maximum dB range: unlimited Maximum STFT frequencies: unlimited

**Spectrum Result**

☒ Frequency is X and time is Z ☐ One data set  
☐ Frequency is Z and time is X ☐ Separate data sets for X and Y components  
☐ Frequency is X and speed is Z Speed:   
☐ Frequency is Z and speed is X

OK Cancel Apply

7. Next, click **OK** to close the dialog box. FlexPro will now add a new object called **DataSpectrum**, which will generate a list with seven spectra.

Now for the second analysis step: determining the amplitude maxima.

1. Select **DataSpectrum** from the Object List. From the FlexPro **New** menu, select **Statistics** and from within that select **Maximum**.
2. Close the **Select List Items** dialog box again by clicking **OK**.
3. In the **Statistical Quantity Properties** dialog box, **Quantities for rows** is already selected. Do not change the default settings. Just click **OK** to close the dialog box.

The key is to understand the **Quantities for rows** setting. This means that for each row in the original data set - which is two-dimensional - a maximum is to be generated. You will understand what this means if you look at a time-frequency spectrum in the data view:

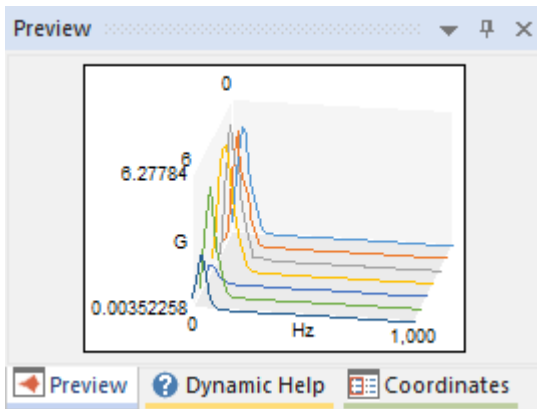
Signal series with 53 signals with 65 64-bit floating point points and Z component [G, Hz, s]

	X	0	1	
Z		0.03175	0.06375	
0	0	0.0491186719773355	0.0136118087622605	0.009
1	15.625	0.0621384972985364	0.0259508283753492	0.0
2	31.25	0.0130177010738318	0.0184550686703916	0.0
3	46.875	0.00789654329638866	0.0140438154900961	0.0
4	62.5	0.00626666629725547	0.0100699034336851	0.008
5	78.125	0.00823740799761621	0.00546244392628491	0.003
6	93.75	0.0109351274390761	0.00348329787502796	0.00
7	109.375	0.0118198591960188	0.00500681715051603	0.0
8	125	0.030329705200335	0.0447834884559392	0.0
9	140.625	0.0459429937596576	0.0713999060168355	0.
10	156.25	0.029411506306331	0.0429666981861308	0.04
11	171.875	0.0081663941476544	0.0130183167033338	0.007
12	187.5	0.00435895819895626	0.00631064522805788	0.004
13	203.125	0.00285970579519442	0.00314998962900413	0.008
14	218.75	0.00296915999597009	0.000463747154015199	0.008
15	234.375	0.00223209117060608	0.00190454242310231	0.004
16	250	0.0033317150396153	0.00141863881769519	0.002
17	265.625	0.00447041666340823	0.00505731957438095	0.00
18	281.25	0.004705881630701	0.00540384754606888	0.008

The Z component with the times is in the top row. The frequencies are in the left-hand column and the amplitudes are in the matrix. Now if a maximum is to be generated for each row, then the result is a signal with the maximum spectral amplitude across the frequency, i.e. a maximum is generated for each row of the matrix.

The **DataSpectrumMaximum** object generates a list with seven of these types of signals:





To obtain the first result of the analysis (the maximum spectral amplitude in the acceleration signal), you only have to generate the maximum one more time.

4. Select **DataSpectrumMaximum** from the Object List. From the FlexPro **New** menu, select **Statistics** and from within that select **Maximum**.
5. Close the **Select List Items** dialog box again by clicking **OK**.
6. In the **Statistical Quantity Properties** do not change the default settings. Just click **OK**.
7. Press F2 and rename the newly created object to **SpectralAmplitude**. The object generates a list with seven scalar values.

In the third step in the analysis, you will determine the frequencies at which the maxima occur. The **Event Isolation** analysis object is ideal for this purpose. You can use it to search for the maxima in the signals that generate the **DataSpectrumMaximum** and output the frequency at which they occur:

8. Select **DataSpectrumMaximum** from the Object List again. From the FlexPro **New** menu, select **Filters** and from within that select **Event Isolation**.
9. Close the **Select List Items** dialog box again by clicking **OK**.
10. In the **Event Isolation Properties** dialog box, double-click on the preset event.
11. In the **Event** dialog box, select the **event** called **Global Extrema**.
12. Leave **Positive** as the **Orientation**. This will ensure that only the global maximum will be found. Next, click **OK** to close the dialog box.
13. In the **Event Isolation Properties** dialog box, select **X value of next event** as the **result**. The signals that provide the **DataSpectrumMaximum** have the frequency you are looking for in their X component.

14. Next, click **OK** to close the dialog box.

15. Press F2 and rename the newly created object to **Frequency**. The object generates a list with seven scalar values.

### Displaying events as tables

1. Select the two results **SpectralAmplitude** and **Frequency** in the Object List. From the FlexPro **New** menu, select **Table**.
2. **SpectralAmplitude** generates a list, which is why the **Select List Items** dialog box appears. Select the option **Convert the list into a data series**. This will ensure that the seven scalar values are displayed one after another in a single column and not side by side in seven columns.
3. Next, click **OK** to close the dialog box.
4. **Frequency** also generates a list, which is why the **Select List Items** dialog box reappears. As before, select the option **Convert the list into a data series** and then click **OK** to close the dialog box.
5. Leave the default selection in the wizard and click **Finish**.

The new table should now look like this:

No.	Spectral Amplitude [G]	Frequency [Hz]
1	0.886905	31.25
2	5.11392	46.875
3	2.4986	31.25
4	5.90233	62.5
5	6.27784	54.6875
6	5.40521	54.6875
7	4.9285	46.875

To improve the legibility of the table, the measurement name and the serial numbers of the measured unit under test need to be added as columns. You can access these attributes using additional data queries:

1. Select the **DataQuery** object in the Object List and choose **DataQuery** from the FlexPro **New** menu.
2. In the **Data Query Properties** dialog box, select **Parent folder** under **Result**. Since the acceleration signals are in the folders labeled with the measurement names, this will provide the information you are looking for.
3. Next, click **OK** to close the dialog box.

4. Press F2 and rename the newly created object to **MeasurementName**. The object generates a data series with seven strings.
5. Drag the mouse to move the **MeasurementName** object from the Object List to the second column of the table and drop it there.
6. Widen the table border so that all columns are displayed completely.

The new table should now look like this:

No.	Measurement Name	Spectral Amplitude [G]	Frequency [Hz]
1	Measurement 5	0.886905	31.25
2	Measurement 6	5.11392	46.875
3	Measurement 7	2.4986	31.25
4	Measurement 4	5.90233	62.5
5	Measurement 3	6.27784	54.6875
6	Measurement 2	5.40521	54.6875
7	Measurement 1	4.9285	46.875

Now only the serial numbers are missing. These are in the acceleration signals as the text parameter "SN".

7. Select the **DataQuery** object in the Object List again and choose **DataQuery** from the FlexPro **New** menu.
8. In the **Data Query Properties** dialog box, select **Parameter text** under **Result**. Under **Parameters** next to it, select **SN**.
9. Next, click **OK** to close the dialog box.
10. Press F2 and rename the newly created object to **SerialNumber**. The object generates a data series with seven strings.
11. Add this result to the table using drag-and-drop as well.

The finished table should now look like this:

No.	Measurement Name	Serial Number	Spectral Amplitude [G]	Frequency [Hz]
1	Measurement 5	00376	0.886905	31.25
2	Measurement 6	06723	5.11392	46.875
3	Measurement 7	11332	2.4986	31.25
4	Measurement 4	11578	5.90233	62.5
5	Measurement 3	26717	6.27784	54.6875
6	Measurement 2	47201	5.40521	54.6875
7	Measurement 1	53769	4.9285	46.875

---

**Note:** If the column arrangement is not correct yet, you can resort them in the table using drag-and-drop.

---

## Sorting the Results

It is clear that the order of the information is not correct yet. This is because the data query provides the data in random order.

Do the following to sort the results by measurement name:

1. Double-click on **DataQuery** in the Object List to open the **Data Query Properties** dialog box.
2. Under Sort, select **Parent folder**.
3. Next, click **OK** to close the dialog box.

If in the FlexPro **Run** menu you selected the option **Automatic Update**, the table will be refreshed immediately. Otherwise, you will have to click **Update** in this menu.

## Excluding outliers from the analysis

The amplitude for Measurement 5 is significantly below that of the rest of the measurements. Refining the data query lets you exclude this data set from the analysis.

1. Double-click on the **DataQuery** object in the Object List.
2. Click on **Add Search Criterion** and select **Maximum** for the **Search element**. For the **Search condition**, select **Greater than or equal to** and for the **Value** enter 2. To close the dialog box, click **OK**.
3. Test your data query by clicking the **Run data query** button under **Result**. The message "6 item(s) found" should now appear on the left.
4. To close the Properties dialog box, click **OK**.

5. If the table does not update automatically, click in the table and then press F12 so that it will refresh.

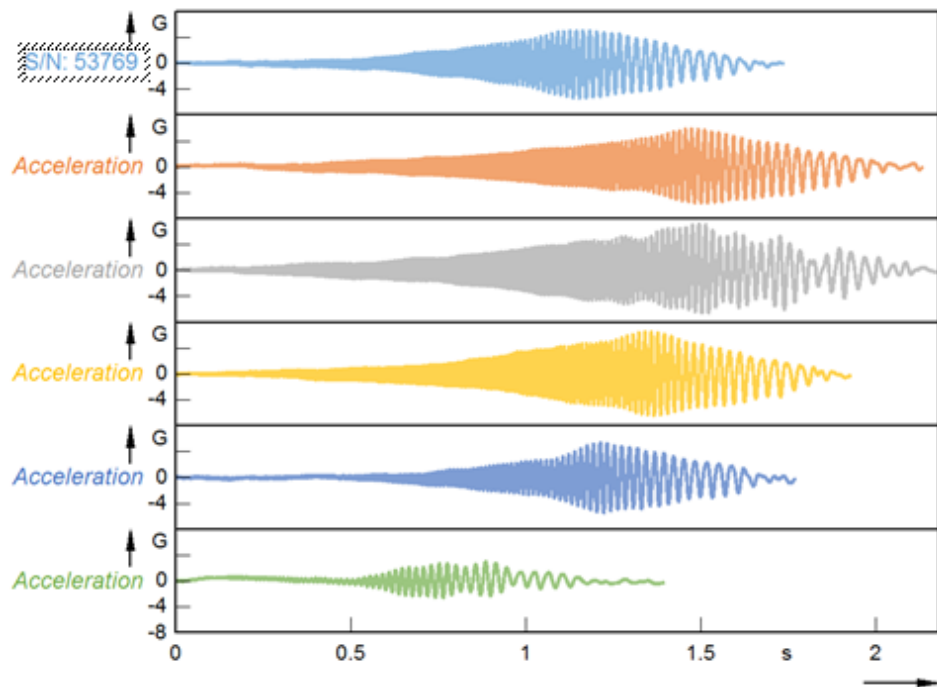
Measurement 5 now no longer appears in the table.

### **Displaying the acceleration signals as a diagram**

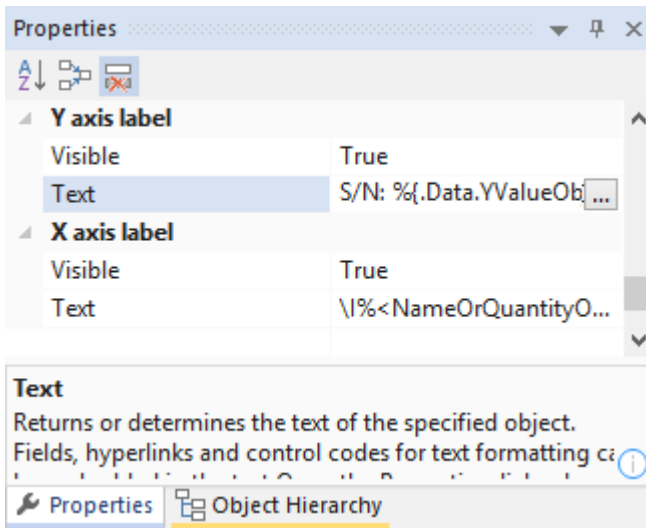
1. Select the **Data** object in the Object List.
2. Click **Insert diagram** on the FlexPro toolbar, or select **Diagram** from the **New** menu.
3. Select **Lines stacked** for the subtype and click **Finish**.
4. The diagram is created and opened. Use the **Toggle Cursors** command from the **Cursors** menu to switch off the cursors.

The data set name is used as the axis label by default. In this case it is not very descriptive because all of the data sets have the same name. Do the following to use the serial number for the name instead:

1. Double-click on the axis label of the top curve to open the Properties dialog box.
2. Under **Y Axis Labeling**, delete the existing text.
3. Now choose **Parameters** from the **Fields** drop-down menu.
4. In the field listed, replace the <Name> placeholder with SN and type S/N: before the inserted placeholder. The text should now look like this: S/N: %{.Data.YValueObject.Parameters("SN")}
5. Before closing the dialog box by clicking on **OK**, select the text and copy it to the Clipboard using CTRL+C. You will need to use this text for the remaining curves as well. The following serial number will be output on the axis of the top curve:



6. Select the axis labels of the other curves by first selecting the label of the second curve, then holding down the CTRL key while clicking on the remaining labels.
7. Now scroll down in the Properties window until you see the **Y Axis Labeling** dialog box.
8. Delete the **Text** field and paste the new axis label from the Clipboard using CTRL+V.



## Analyzing Data on the Hard Drive

If you have a license for the **Data Explorer option**, you can run queries directly on data on your hard drive.

### Setting up indexing

1. If Data Explorer does not appear, press CTRL+ALT+X or select **Data Explorer** from the **Window** menu of the **View** menu to open the window.
2. Click **Options** in the toolbar of the Data Explorer window to view the **Indexing** tab of the **Option** dialog box.
3. For the **Mode**, select **Always index**.
4. Under **File Types To Index** select the option **Index all supported file types**.
5. Under **Locations To Index** make sure that C:\Users\Public is entered and highlighted. Under **Subfolders** make sure **Yes** is selected. FlexPro example files are in this folder.
6. Now click on the **Details** button at the bottom right to view the file indexing status.
7. Wait, if necessary, until the files Measurement 1.fpf through Measurement 7.fpf have been indexed and then close both dialog boxes by clicking **OK**.

**Note:** If the files are not displayed and you are not sure if they were indexed correctly, you should click the **Delete and create new index** button in the **Options** dialog box and then open the status window again.

---

### Displaying the acceleration signals in Data Explorer

1. Click the **Search** button in Data Explorer to open the **Select / Edit Search Filter** dialog box.
  2. Under **Search filter in**, select **This project database**. This allows you to use the search filter later for other projects as well.
  3. Now click **Add New Search Filter** and enter **AccelerationSignals** for the name.
- Now the seven acceleration signals will be visible in Data Explorer.

### Useful information about Data Explorer

- Use the **Change View** button in the toolbar of the Data Explorer window to toggle between the list and tree views.
- If you select a data set, it will appear in the Preview and its header information will appear in the Properties window.
- To import one or more data sets, select them in Data Explorer and drag them with your mouse to the Object List. If you press and hold CTRL and SHIFT at the same time, links will be added to the project database.
- Alternatively, you can also copy or link them via the Data Explorer context menu. To do this, right-click with your mouse on the highlighted item(s).
- If you click on the arrow next to the Data Explorer **Search** button, a menu will appear in which you can select a search filter.
- In the project database or template database you can save as many search filters as you need. In Data Explorer itself you can add a standard search filter that you can select from the menu using **Find....**
- Search filters that you add to the project database or template database can also be used in data queries.

### Setting up the data query for external data

The previously added DataQuery object will access the imported measurements 1 through 7 and so it should now be changed to access the data on the hard drive directly.

1. Double-click on the **DataQuery** object to open its Properties window.
2. Under **Search in**, select **Indexed files**.
3. Now click the button under **Search criteria** called **Load Search Criteria**.



4. In the **Search Filters** dialog box, select **This project database** under **Search filter in**.
5. The search filter **AccelerationSignals** should already be selected. If not, select it from the **Name** list.
6. Next, click **OK** to close the dialog box so that you can copy the search filter search criteria to the **DataQuery** object.
7. Click **Run data query** to verify that seven items are found.
8. The criterion for excluding outliers is still missing. You will therefore need to add the **Greater than or equal to 2** criterion to the list.
9. The linked data queries have not yet been switched to the external data source. Select the objects **Data**, **MeasurementName** and **SerialNumber** in the Object List, scroll down the FlexPro Properties window until you see **Search in** and select **In indexed files**.

---

**Note:** Using the object key, which the **DataQuery** object provides, FlexPro cannot detect whether the key relates to data in the project database or external data. You will therefore always have to make sure that the same source is set in the linked queries as in the query that provides the key.

---

Now if you update the previously created diagram and table, they should provide the exact same results as that of the query for imported data. The measurement folders **Measurement 1** through **Measurement 7** are now no longer required, so you can delete them from the project database.

## 2.4 Customizing FlexPro

The tutorials for customizing FlexPro cover the creation of presentation and analysis templates as well as expansion of the FPScript function libraries to include custom functions.

You can find them in FlexPro Online Help.

## 3 Data Management

### 3.1 Project Database

FlexPro stores all objects that you create or import in a project database. The inner structure of a FlexPro project database resembles the file system on a hard drive. Instead of files on the hard drive, a FlexPro project database contains data sets, formulas, presentation objects, worksheets and documents that you need for your analysis. Just like on a hard drive, you can build up a hierarchy of folders in a FlexPro project database to organize your FlexPro objects.

#### How Project Databases are Stored

When you store a project database, you are offered two options for saving the project database. Your work with FlexPro does not depend on how you have set up your project databases. The distinction is made for purely technical reasons.

- **In a single file**

This is the default storage method for a new project database. All FlexPro objects are combined into one file. The advantage of this format is that it is easier to use because only a single file is set up on the hard drive. If you would like to back up your project database or transfer it to another computer, you only have to copy a single file, e.g. **ProjectDatabase.FPD**. One disadvantage of this format is the slightly slower access speed, especially when you are working with large amounts of data. A further disadvantage is the higher risk of data loss in the event that the inner structure of the project database file becomes corrupted, such as due to a system failure while saving the file. FlexPro can recognize and delete defective objects within the file. In rare instances, however, it may no longer be possible to read the project database.

- **In several folders and files**

This is an alternative storage option. In this case, the folder hierarchy which you build up in your project database is associated with folders on your hard drive. When you create the **Project Database** project database, in addition to the **ProjectDatabase.FPD** file, two subfolders called **ProjectDatabase.DB** and **ProjectDatabase.TMP** are added to the same folder. The project database folder hierarchy is stored under the **ProjectDatabase.DB** hard drive folder. All FlexPro objects are saved at the corresponding positions within the hierarchy as separate files. The **ProjectDatabase.TMP** folder is used to store temporary files and only exists while the **ProjectDatabase.FPD** is open.

If you want to make a backup of your project database, you will have to make a copy of the file **ProjectDatabase.FPD** as well as of the **ProjectDatabase.DB**

folder.

The folder-oriented storage system is recommended when you are working with very large amounts of data. You can increase FlexPro's performance by using this type of storage. In addition, there are no limitations in this format with regard to the total size of the project database and the maximum number of folders within it.

---

**Note:** If possible, you should not make any changes to FlexPro objects under the DB folder. All relevant operations are available in FlexPro.

---

### **Project Database Storage Options**

You can determine whether calculated images of presentation objects and calculated results of formulas should be stored in the project database by setting this in the project database Properties dialog box. Elements that you store in the project database will not have to be recalculated the next time you open the project database. On the other hand, in certain circumstances considerably more space may be used on the hard drive when saving the data, particularly when saving the results of formulas.

### **Different Modes for Opening Project Databases**

You can choose from three different modes to open a project database:

- **Open backup for editing**

When a project database is opened, a backup copy is created. All further work is done in this copy. When the work is saved, the changes to the copy are transferred to the original database. The copy is deleted when the project database is closed. Please note that you can only use this mode if you have sufficient storage space. **Important:** backup copies of externally stored Excel workbooks that are linked in the project database will not be created.

This is the default mode for opening project databases.

- **Open as read-only**

Use this mode to open your project database if you want to be absolutely certain that no data in the project database can be changed inadvertently. Project databases that are write-protected, such as those on a CD, are automatically opened using this mode.

- **Open for direct editing**

Use this mode if you would like to edit the project database directly or if you do not have enough space available to work with a backup copy. All changes are carried out directly in the project database without a request for confirmation.

**Multiple users competing for access to one project database**

If a user already has a project database open, additional users can work with the project database at the same time. In this case a copy of the original has to be created. The following table provides an overview of all possible scenarios:

Mode: project data base is already open	Mode: project database is to be opened		
	Backup	Read-only	Direct
With backup copy	Possible, work carried out on copy of original	Possible, work carried out on copy of original	Possible, work carried out on copy of original
Read-Only	Possible, work carried out on copy of original	Possible	Possible, work carried out on copy of original
Direct	Not possible	Not possible	Not possible

Possible conflicts are automatically recognized, and the system asks whether a copy should be created. Please note that any changes made to the copy will have to be transferred manually to the original database once it is accessible.

---

**Note:** Project databases appearing in the list of recently used project databases in the File menu will be opened in the same mode as when they were last opened. A project database will first be closed if it is already open and is then opened again in a different mode. The system will then try to open it in the requested new mode.

---

**Compatibility**

FlexPro is backward compatible with previous versions of FlexPro, which means that you can use project databases that you created with older versions of FlexPro in FlexPro 10. If you open this type of project database in FlexPro 10. It is then converted to the FlexPro 10 format and can no longer be opened in the previous version of FlexPro.

If you purchased FlexPro 10 as an update to a FlexPro 9 license, you can continue to use FlexPro 9. FlexPro 9 also runs using the FlexPro 10 license. If you want to edit a FlexPro 9 project database and you want to ensure that the database can continue to be read in FlexPro 9, then you should edit it in FlexPro 9.

**Data Indexing**

Indexing project database data speeds up searches in project databases and provides for flexible searching for data using the **DataQuery** object.

## Template Databases

FlexPro template databases are used to store presentation and analysis templates, macros, units and FScript functions. The following are the project databases that FlexPro keeps open or opens briefly in order to import settings:

- **Personal template database**

This is assigned to your Windows user account and is stored on your computer in your Windows user profile.

- **Shared template databases (FlexPro Professional, Developer Suite only)**

As a FlexPro Professional and/or Developer Suite user, you can set up a list of shared project databases that are usually located in network folders. These template databases are opened in FlexPro as read-only and only briefly during normal operation so that multiple users can use them at the same time. Exclusive access applies only when you use the **Organizer** dialog box. You cannot store macros in shared project databases. Use them to share analyses, unit tables and FScript function libraries within a team.

## Working with Project Databases

### Creating a New Project Database

#### To create a new project database with a backup copy:

- Select **New Project Database** from the **File** menu.

---

**Note:** Project databases that you create this way are then stored in a temporary folder on your hard drive.

---

#### To create a new project database for direct editing:

1. Select **New Project Database** from the **File** menu to create a new project database.
2. Select **Save Project Database** from the **File** menu.
3. In the dialog box, select the folder, enter a name for the project database and select the storage format, such as **In a single file**.
4. Click on **Close Project Database** from the **File** menu.
5. Reopen the project database using **Open Project Database** from the **File** menu and select the mode **Open for direct editing**.

## Searching through a Project Database

You can search the active project database or parts of it for objects that fulfill particular criteria. Do the following:

1. In the Folders window, select the folder that you want to search.
2. Select **Finding** from the **Edit** menu. The **Search in Project Database** window appears.
3. Select the option **Include subfolders in search** if subfolders of the selected folder should also be searched.
4. Select the option **Output in Find Results 2** if the objects found should be output to the **Find Results 2** window instead of the **Find Results 1** window.
5. Select the option **Use search index** if you want to speed up the search process. In this case, objects not yet calculated will be ignored during the search. To do this, Data Indexing must be enabled for the project database. The search process may be slow if the **Use search index** option is not used. However, objects not yet calculated will be updated during the search.
6. The next step is to determine the criteria that the objects searched for should fulfill. You can combine multiple criteria by clicking on **More** and setting the desired logical operation. The search criterion can be combined with a Boolean AND or OR, where AND takes precedence over OR. The individual search items and search operations are listed below. The input search value is also covered below.
7. Click on **New Search**. The **Find Results** window is first cleared and then the objects found are displayed. If you want to start a new search while keeping the objects currently displayed in the **Find Results** window, then click **Search** instead of **New Search**.
8. You can cancel the search at any time by clicking on **Stop**.

The editing options in the **Find Results** window where the objects found are displayed mostly match those of the object list.

## Search Item

The following table lists all search items supported by FlexPro. In the right-hand column you can see whether a search item is available for searching in the project database, for index-based searching in the project database, or for index-based searching in external files:

Criterion	Description	Available in
Absolute end time	Date and time of last X value. The data set must either have	

Criterion	Description	Available in
	an X component or it must be selected as the X component itself.	
Absolute start time	Date and time of first X value. The data set must either have an X component or it must be selected as the X component itself.	
Sampling rate	The sampling rate of an equidistant data set or of its Y component.	
Sampling rate of X component	The sampling rate of a data set's X component.	
Sampling rate of Z component	The sampling rate of a data set's Z component.	
Date modified	Date and time of the last modification of an object.	
Number of dimensions	The number of dimensions for a data set, e.g. 2 for a data matrix.	
Number of columns	The number of columns in a two-dimensional data set.	
Number of rows	The number of rows in a data set. In the case of a one-dimensional data set, this is its number of values.	
Author	The name of the data set author.	
Upper range limit	The upper range limit of a data set or of its Y component.	
Upper range limit of X component	The data upper range limit of a data set's X component.	
Upper range limit of Z component	The data upper range limit of a data set's Z component.	
Lower range limit	The data lower range limit of a data set or of its Y component.	
Lower range limit of X component	The data lower range limit of	

Criterion	Description	Available in
	a data set's X component.	
Lower range unit of Z component	The data lower range limit of a data set's Z component.	
File extension	The file name extension of the file from which the data set originates, such as ".fpf" for FlexPro text data.	File index only
File name	The name of the file from which the data set originates.	File index only
File type	The type of file from which the data set originates, as it is displayed in the FlexPro Import dialog box, e.g. "FlexPro Text Data (*.fpf)".	File index only
Data structure	The data structure of a data set.	
Data type	The data type of a data set or of its Y component.	
Data type of X component	The data type of a data set's X component.	
Data type of Z component	The data type of a data set's Z component.	
Digital	The "Digital" attribute of a data set or of its Y component. (True, if the data set contains zeros and ones.)	
Digital X component	The "Digital" attribute of a data set's X component.	
Digital Z component	The "Digital" attribute of a data set's Z component.	
Unit	The unit of a data set or of its Y component.	
Unit of the X component	The unit of a data set's X component.	
Unit of the Z component	The unit of a data set's Z component.	
Scalar value (numeric)	The numeric scalar value of a data set.	



Criterion	Description	Available in
Scalar value (string)	The scalar string value of a data set.	
Creation date	Date and time of the creation of an object.	File index only
Formula	The FScript code of a formula or of an object derived from a formula.	
Quantity	The name of the physical quantity of a data set or of its Y component.	
Quantity of X component	The name of the physical quantity of a data set's X component.	
Quantity of Z component	The name of the physical quantity of a data set's Z component.	
Grandparent folder	For an object in the project database, this is the name of the folder containing the folder in which the object is located. For an external data set, this is the name of the folder in which the file that contains the data set is located.	
Origin	The origin of a data set. Usually this is the path name of the file from which it originates.	
Hyperlink	The hyperlink assigned to an object.	
Increment	The increment of an equidistant data set or of its Y component.	
X component increment	The increment of a data set's equidistant X component.	
Z component increment	The increment of a data set's equidistant Z component.	
Internal path	The path of an object inside a	

Criterion	Description	Available in
	project database.	
Category	The object category assigned with the Categorize Object List command.	
Comments	The comments of an object or of a data set's Y component.	
Comments of X component	The comments of a data set's X component.	
Comments of Z component	The comments of a data set's Z component.	
Machine	The host name of the computer on which the file is stored from which a data set originates.	File index only
Maximum	The maximum of a data set or of its Y component.	
Maximum of X component	The maximum of a data set's X component.	
Maximum of Z component	The maximum of a data set's Z component.	
Minimum	The minimum of a data set or of its Y component.	
Minimum of X component	The minimum of a data set's X component.	
Minimum of Z component	The minimum of a data set's Z component.	
Mean	The mean value of a data set or of its Y component.	
Mean value of X component	The mean value of a data set's X component.	
Mean value of Z component	The mean value of a data set's Z component.	
Name	The object name.	
Object text	Any text appearing in an object.	Search in database only; no index
Object type	The object type.	
Folder path	The path of the folder on the	File index only

Criterion	Description	Available in
	hard drive on which the file is stored from which a data set originates.	
Parameter name	The name of a parameter entered on an object's <b>Parameters</b> tab.	
Parameter name and text	The name and text of a string parameter. Both attributes must match.	Index-based search only
Parameter name and value	The name and value of a numeric parameter. Both attributes must match.	Index-based search only
Parameter text	The text of a string type parameter.	
Parameter value	The value of a numeric parameter.	
Parameter unit*	The physical unit of a numeric parameter.	
Relative end time	The last X value of a data set. The data set must either have an X component or it must be selected as the X component itself.	
Relative start time	The first X value of a data set. The data set must either have an X component or it must be selected as the X component itself.	
Standard deviation	The standard deviation of a data set or of its Y component.	
Standard deviation of X component	The standard deviation of a data set's X component.	
Standard deviation of Z component	The standard deviation of a data set's Z component.	
Void values	The "Contains void values" attribute of a data set or of its Y component.	
Void values in the X component	The "Contains void values"	

Criterion	Description	Available in
	attribute of a data set's X component.	
Void values in the Z component	The "Contains void values" attribute of a data set's Z component.	
Parent folder	For an object in the project database, this is the name of the folder that contains the object. For an external data set, this is the name of the file that contains the data set (without the file extension).	
Timestamp	The timestamp of a data set or of its Y component.	
Timestamp of X component	The timestamp of a data set's X component.	
Timestamp of Z component	The timestamp of a data set's Z component.	

\* When searching for the unit, the search is case sensitive. When searching in other text, however, case does not matter.

### Search operations

Depending on the search criterion data type, you can choose one of the following search operations:

Data type	Operation	Search for object if...	Available in
Text	contains	The specified text appears in the searched text.	
	does not contain	The specified text does not appear in the searched text.	
	is	The specified text is identical to the text in the searched text.	
	begins with	The text that is searched starts with the specified search text.	
	ends with	The text that is searched ends with the specified	

Data type	Operation	Search for object if...	Available in
		search text.	
	matches	The text to search matches the specified search pattern. Details, see below.	Search in database only; no index
Numeric	equal to	The value to find is identical to the reference value.	
	less than	The reference value is smaller than the value to find.	
	less than or equal to	The reference value is smaller than or equal to the value to find.	
	greater than	The reference value is greater than the value to find.	
	greater than or equal to	The reference value is greater than or equal to the value to find.	
Date & Time	equal to	The reference date matches that of the date to find. The time is ignored in this comparison.	
	older	The date and time occur after the date & time specified for the search.	
	newer	The date and time occur before the date & time specified for the search.	
Object	is	The searched object matches the selected type.	
	is not	The searched object does not match the selected type.	
Attribute	true	If the attribute to check is present.	
	not true	If the attribute to check is present.	

## Entering search values

You can enter numeric values as floating point, time span or calendar time, e.g.

123

-1.25

1.3E-10

12:13:14.56

4:12:13:14

23.1.2010 12:13:14.56

23.2.2010

2/23/2010 12:13:14.56

You can also enter complex values:

1+3i

(1 ; 3)

FlexPro forms its absolute value and uses it for the search.

You can also calculate the value using an FScript expression. Start the formula with '=':

= 0.5 \* PI

Please note that in FScript code the period is always used as the decimal separator.

## Recognizing templates when searching for text

The **matches** search criterion helps you search for text patterns. The text pattern is defined here as what is called a regular expression. All text is found that matches the given regular expression. In regular expressions, special characters and sequences are used to represent a text pattern. The following table describes these characters and sequences and provides some examples.

Character	Description
^	Finds the beginning of the string.
\$	Finds the end of the string.
*	Finds zero or more occurrences of the preceding character. For example, "zo*" finds "z" or "zoo".
+	Finds one or more occurrences of the preceding character. For example, "zo+" finds "zo", but not "z".
?	Finds the preceding character zero times or once. For example, "b?ig?" finds "ig" in "signal".

Character	Description
.	Finds every single character.
x y	Finds either x or y. For example, "z wood" finds "z" or "wood". "(z w)oo" finds "zoo" or "wood".
{n}	n is a non-negative integer. Finds exactly n times. For example, "o{2}" does not find the "o" in "Bob", but it does find the first two instances of "o" in "fooooo".
{n,}	n is a non-negative integer. Finds at least n times. For example, "o{2,}" does not find the "o" in "Bob", but finds all instances of "o" in "fooooo". "o{1,}" is equivalent to "o+". "o{0,}" is equivalent to "o*".
{n,m}	m and n are non-negative integers. Finds at least n and at most m times. For example, "o{1,3}" finds the first three instances of "o" in "fooooo". "o{0,1}" is equivalent to "o?".
[xyz]	A group of characters. Finds every one of the characters included. For example, "[abc]" finds the "a" in "signal".
[^xyz]	A group of excluded characters. Finds every character not included. For example, "[^abc]" finds the "s" in "signal".
[a-z]	A range of characters. Finds every character in the range. For example, "[a-z]" finds every alphabetical character from "a" to "z".
[^m-z]	A range of excluded characters. Finds every character not in this range. For example, "[m-z]" finds every character that does not lie between "m" and "z".
\b	Finds a word limit, i.e. the position between a word and a space. For example, "a\b" finds the "a" in "signal", but not the "a" in "signals".
\B	Finds a non-word limit. "ea*r\B" finds "ear" in "never early".
\d	Finds a number. Equivalent to [0-9].
\D	Finds a non-numeric character. Equivalent to [^0-9].
\s	Finds a space character. Equivalent to "[ ]".
\S	Finds everything but space characters. Equivalent to "[^ ]".
\w	Finds every word character including underline. Equivalent to "[A-Za-z0-9_]".
\W	Finds every character that does not belong to a word. Equivalent to "[^A-Za-z0-9_]".

---

**Note:** The search only takes into account text that is found for which the regular expression is a pattern for the complete text. "Signal+", for instance, finds the object with the name "Signal1", but not the object "Signal20". The regular expression "Signal+" only looks for the string "Signal2" in "Signal20".

---

### **Sending a Project Database via E-mail**

- You can send the current project database as an e-mail attachment by using **Send Project Database as E-Mail** from the **File** menu.

---

**Note:** You have to have an e-mail client such as Microsoft Outlook installed on your computer in order to use this command.

---

### **Defragmenting a Project Database**

- **Defragment Project Database** in the **File** menu allows you to defragment a project database as long as it is not currently opened in FlexPro or marked as read-only.

---

**Note:** When defragmenting, free space is removed from the file and fragmented data is consolidated. A defragmented project database is usually more compact and allows for faster data access.

---

### **Saving the Workspace**

**To save the layout of the database window currently open in the workspace:**

- Select **Save Workspace** from the **Window** menu.

**To save the layout of the window currently open in the workspace automatically every time the database is saved:**

- Select **Save Workspace Automatically** from the **Window** menu.

---

**Note:** When opening a database, the workspace last saved will be restored automatically.

---

### **Optimizing FlexPro for Large Volumes of Data**

FlexPro combines a high level of ease of use with high performance in a unique way and is ideal for a wide range of applications. In some areas, however, a higher level of ease of use and robustness can only be achieved when there is a loss of data throughput. FlexPro solves this problem by offering you different settings that make it possible for you to optimize the application for ease of use and data throughput. FlexPro's default



settings are optimized for the highest level of ease of use and robustness. Please note the following if you want to optimize FlexPro for large volumes of data or for high data throughput.

### **Project Database Format and Operating Mode**

The standard way that FlexPro operates is to work with a temporary copy of the FlexPro project database and to save it first where you specify by using the **Save Project Database** command. When working with large volumes of data, you should create the database and save it in the **In several folders and files** at the preferred location on your hard drive before actually starting to work with it, then open it in **For direct editing** mode:

1. Start FlexPro to obtain an empty project database.
2. Select **Save As** from the **File** menu.
3. In the **Save As** dialog box, select the hard drive folder in which the project database is to be saved.

---

**Note:** You should save the project database to a local hard drive and not in a folder on your network.

---

4. Now enter the desired name in the **File Name** field.
5. Select **In several folders and files as the storage format** and then click **Save**.
6. Now click on **Open Project Database** from the **File** menu.
7. In the **Open** dialog box, select the project database that you just saved.

---

**Note:** The project database file extension is **.fpd**. In the same folder is a subfolder of the same name with the file extension **.db**. This folder will later contain all objects stored as files.

---

8. Select the mode **Open for direct editing**.

The project database is now saved and can be used.

### **Deactivating Global Link Correction When Renaming Objects**

If you rename an object in a FlexPro project database, FlexPro searches the entire project database for objects that have references to this object in order to correct them. This can take a lot of time if the project database is large.

1. Select **Options** from the **Tools** menu.

2. Click on the **Operation** tab.
3. Uncheck the option **Search the complete project database when looking for references**. FlexPro then searches only folders in which the renamed object is located.

### Deactivating the Preview

FlexPro's Preview option may be deactivated to increase the data throughput. To do this, click on **Close** in the title bar of the preview window or uncheck the **Show Preview Window** option in the **Object List** toolbar.

### Preventing Recalculation of Images and Calculated Results

FlexPro can save images of diagrams and tables as well as calculated results of formulas in the project database. These are immediately available and do not have to be recalculated when you re-open the project database.

1. Select **Options** from the **Tools** menu.
2. In the **Options** dialog box, switch to the **Project Database** tab.
3. Select the options **Save calculated images of presentation objects in the project database** and **Save calculated results of formulas in the project database**.

### File Import

If you want to import data into the FlexPro project database, you should only select the necessary data sets and import them in **Copy** mode. If you are working with equidistantly sampled signals, you should import them as signals.

1. Select **Import** from the **File** menu.
2. In the **Import** dialog box, uncheck **Fast import of complete files** and **Create links** and select **Import as signals**.
3. Click on **Open** to import the data.

### Using Cursors for Large Data Sets

If you would like to use cursors for several channels of a large measurement, for instance, you can either use a single diagram with stacked Y axes or a worksheet object with one plane per diagram. The worksheet provides more options for zooming and scrolling the individual channels independently.

1. Select all of the data sets for which you want to use cursors.

2. Click on **Insert Worksheet** in the **FlexPro Objects** toolbar.
3. Select the desired window layout and click on **Finish**.

The cursors are placed when positioning the mouse on the point closest to the position clicked. To do this, the distances for all XY value pairs for the curve are calculated, which causes a delay when working with large volumes of data. Therefore, you should switch to a mode that accounts for only the selected X position.

1. Right-click with your mouse on the worksheet and select **Properties**.
2. In the **Properties** dialog box, click on the **Cursors** tab.
3. Under **Cursor positioning**, select the option **Use X position**.

When zooming to sections, you should select **Spread X Range Between Cursors** with rubber banding as the preferred method.

1. Click on the right side of the desired section to place the leading cursor at this position.
2. Press the TAB key to activate the origin cursor.
3. Click on the left side of the desired section.
4. Click on **Spread X Range Between Cursors** in the **Cursor** toolbar.

---

**Note:** You should select the smallest section as possible. Subsequent scroll operations will then be faster. To scroll, you should use the arrow on the scrollbar for shorter moves. You should only use the slider on the scrollbar for long jumps on the curve.

---

## Optimizing the Analysis

You should setup and test your analysis smaller data sets and then apply the prepared analysis to a large volume of data. For instance, you can import a relatively small data set, use FlexPro's Reduce function to decimate the data, or create a section of the data using the Index operator.

You can optimize the amount of run-time environment memory on your computer required for the FPScript analysis language.

1. Choose the command **Options** from the **Tools** menu.
2. Click on the **System Settings** toolbar.
3. Under **Maximum size of data sets in memory**, enter a value in megabytes that is higher than the size of the data sets that you are using. If you have, for instance, data sets with 10,000,000 64-bit floating point values, you should enter a value that is at least than 8 megabytes (1 megabyte = 1,048,576 bytes), since 8 bytes are needed for

each 64-bit floating point value.

4. Under **Maximum memory allocation for data sets**, enter a value that equals no more than 70% of the computer's available memory.
5. Close FlexPro.
6. Now open Windows Task Manager by right-clicking on the taskbar and selecting Task Manager.
7. Now restart FlexPro and carry out a typical data analysis.
8. Follow the **page file** usage on the **Performance** tab of the Task Manager. The value should be displayed as the **total** value in the **Physical Memory** field and should not exceed more than 20% of the memory available on your computer.
9. If the memory requirement limit is exceeded significantly, you should set the value in the **Maximum memory allocation for data sets** field back to a lower value.

### Additional Optimization Options

If you have considered all of the points mentioned above, you can increase FlexPro's data throughput considerably. However, before you start analyzing and managing large volumes of data, you should check to see if the amount of data cannot be reduced when you acquire it. You should set the sampling rate for the measurement in accordance to the bandwidth of the signals to be measured and not to the capabilities of the measurement system. If your system has a good anti-aliasing filter, it generally does not make much sense to oversample the signal considerably. FlexPro also has no trouble jointly processing and displaying data with different sampling rates. You can therefore also capture slower signals, e.g., temperatures, with a lower sampling rate than faster signals.

Additional options for optimization are available when you select the appropriate graphic format. For instance, for a 3D data set, a waterfall chart is much faster than a surface chart. You should also always take into account whether the data must be presented in its full resolution or if the data can be decimated in advance.

### Troubleshooting Project Databases

#### **I deleted a lot of data from my project database, but the file size did not shrink.**

The project database has to be defragmented to delete free space from the file.

- Use the command **Defragment Project Database** from the **File** menu after closing the project database.

**Renaming objects in the project database takes a long time.**

If you rename an object in a FlexPro project database, FlexPro searches the entire project database for objects that have references to this object in order to correct them. This can take a lot of time if the project database is large.

1. Select **Options** from the **Tools** menu.
2. Click on the **Operation** tab.
3. Uncheck the option **Search the complete project database when looking for references**. FlexPro then searches only folders in which the renamed object is located.

**FlexPro uses too much memory.**

You can optimize the memory requirements for FlexPro and the runtime environment for the FScript analysis language for your computer.

1. Select **Options** from the **Tools** menu.
2. Click on the **System Settings** tab.
3. Under **Maximum memory allocation for data sets**, enter a value that equals no more than 70% of the computer's available memory.
4. Under **Maximum size of data sets in memory** enter a smaller value. Usually, there are several small and some large data sets in a project database. The value that you set here should be somewhere in the middle. Note that the size of a data set in bytes equals in the number of values multiplied by the number of bytes per value. A 64-bit floating point value takes up 8 bytes, for instance.
5. Under **Maximum number of folders in memory** enter a smaller value. Try out different values to find the setting that provides optimum performance. Lowering the value has effect only if there is no connection in the form of object references between the folders in the project database.
6. Under **Maximum count of loaded preview images** enter a smaller value. A preview image can take up to 120 KB of memory.

**Upon reopening a database, calculations already performed are repeated.**

You can make sure that FlexPro stores all calculated results in the project database so that they will not have to be recalculated again during the next session.

1. Select **Properties** from the **File** menu.
2. Select the options **Save calculated images of presentation objects in the**

**project database** and **Save calculated results of formulas in the project database**.

---

**Note:** FPScript formulas that provide a vague result, such as functions containing a user prompt or use the Noise function, are recalculated at each update and are not stored in the database.

---

### **The project database requires a considerable amount of hard drive space.**

You can reduce the amount of hard drive space required by project databases by not saving optional elements on the hard drive.

3. Select **Properties** from the **File** menu.
4. Uncheck the options **Save calculated images of presentation objects in the project database** and **Save calculated results of formulas in the project database**.

### **Setting Up a Shared Template Database**

---

**Note:** Only FlexPro Professional and FlexPro Developer Suite support shared template databases.

---

1. In the FlexPro **Tools** menu, select **Organizer**.
2. In the **Organizer** dialog box, click on the **Templates** tab.
3. Next, click on the button **Shared Template Databases**.
4. Click on **Insert template database** to create a new entry.
5. Now enter the full pathnames to the template database in the field or click on the button after the field to select a database.

### **Setting the Save Options**

#### **To change the save options for the active database:**

1. Select **Properties** from the **File** menu.
2. On the **General** tab of the Properties dialog box, select the desired options.

**To set the save options for newly created databases:**

1. Select **Options** from the **Tools** menu.
2. In the **Options** dialog box, switch to the **Database** tab.
3. Select the relevant options under **Default settings for newly created databases**.

**Setting Up Indexing of Project Databases****Changing Indexing for the Active Database**

1. Select **Properties** from the **File** menu.
2. On the **Indexing** tab of the Properties dialog box, select the desired options.

**Setting Up Indexing for Newly Created Databases**

3. Select **Options** from the **Tools** menu.
4. In the **Options** dialog box, switch to the **Project Database** tab.
5. Select the option **Activate data query indexing** under **Default settings for newly created databases**.

**Exporting Project Databases as a Web in HTML Format**

Using HTML Export, you can publish your evaluations as an HTML web within minutes. The HTML web created can then be viewed with any web browser such as Microsoft Internet Explorer or Netscape Navigator. You can export individual objects, folders and subfolders or entire project databases. Several exports can be brought together in a single web.

For each object to be exported, FlexPro generates one HTML page, or in the case of multi-page objects and large databases, several HTML pages. FlexPro can convert tables, text and data sets completely into HTML code. Other objects, such as diagrams and documents, are incorporated into the HTML pages as PNG graphics.

FlexPro can also include a table of contents in the web. Even a hierarchy view such as that in Windows Explorer is possible. This can be integrated as a Java Applet, JavaScript Applet or as an ActiveX control.

**HTML**

HTML (HyperText Markup Language) is the standard "markup language" used for

documents on the World Wide Web. HTML is developed by the World Wide Web Consortium. The HTML programming language uses tags to specify how page elements such as text and graphics are to be displayed in a web browser and how a web browser is to react to user actions such as activating hyperlinks via the keyboard or mouse. Most web browsers, in particular Microsoft Internet Explorer and Netscape Navigator, also recognize HTML tags, which are not yet part of the current standard. During export into HTML format, FlexPro automatically generates HTML files. Therefore, knowledge of the HTML programming language is not required.

## HTML Templates

FlexPro generates the HTML pages for the individual objects based on an HTML template file. For each object type a separate template exists that you can customize. Fields in the template determine which object elements are to be integrated where.

The HTML templates are located in a subfolder within the FlexPro program folder. For normal installation, the path name should be C:\Program Files (or Program Files (x86))\Weisang\FlexPro 10.0\ENU\HTML Templates. To customize the templates to fit your requirements, you should copy the complete folder and change the copied files only. In the second step of the HTML Export Wizard, you can specify the folder from which the templates are to be taken. You can use the following fields in the HTML templates:

Field	Meaning
<!-- %(Activated Folder Comments) -->	Comments on the activated subfolder of the folder containing the object to be exported.
<!-- %(Activated Folder Name) -->	Name of the activated subfolder of the folder containing the object to be exported.
<!-- %(Author) -->	Name of the data object author.
<!-- %(Begin Hyperlink) --> <!-- %(End Hyperlink) -->	The hyperlink specified on the General tab of the Properties dialog box of the object to be exported. The text enclosed by the two fields is highlighted as the hyperlink.
<!-- %(Comments) -->	Comments of the object to be exported.
<!-- %(Comments Or Name) -->	Comments or, if there are no comments, the name of the object to be exported.
<!-- %(Database Name) -->	Name of the project database to be exported.
<!-- %(Database Path) -->	Path of the project database to be exported.
<!-- %(Date) -->	Date of the export.
<!-- %(Folder Comments) -->	Comments of the folder containing the object to be exported.



Field	Meaning
<code>&lt;!-- %(Folder Name) --&gt;</code>	Name of the folder containing the object to be exported.
<code>&lt;!-- %(Name) --&gt;</code>	Name of the object to be exported.
<code>&lt;!-- %(Next Page) --&gt;</code>	For multi-page objects, this is a button to switch to the next page.
<code>&lt;!-- %(Number Of Pages) --&gt;</code>	The number of pages in a document object.
<code>&lt;!-- %(Object) --&gt;</code>	The HTML code of the object to be exported.
<code>&lt;!-- %(Origin) --&gt;</code>	The origin of the data of a data object.
<code>&lt;!-- %(Path) --&gt;</code>	Path of the object to be exported.
<code>&lt;!-- %(Page Number) --&gt;</code>	The current page number of a document object.
<code>&lt;!-- %(Previous Page) --&gt;</code>	For multi-page objects, this is a button to switch to the previous page.
<code>&lt;!-- %(Quantity) --&gt;</code>	The name of the physical quantity of a data object or its Y component.
<code>&lt;!-- %(QuantityX) --&gt;</code>	The name of the physical quantity of a data object's X component.
<code>&lt;!-- %(QuantityZ) --&gt;</code>	The name of the physical quantity of a data object's Z component.
<code>&lt;!-- %(Style) --&gt;</code>	Stylesheet definition that is required for the object.
<code>&lt;!-- %(Table Of Contents) --&gt;</code>	The HTML code of the table of contents.
<code>&lt;!-- %(Time) --&gt;</code>	Time of the export.
<code>&lt;!-- %(Unit) --&gt;</code>	The physical unit of a data object or its Y component.
<code>&lt;!-- %([Unit]) --&gt;</code>	The physical unit of a data object or its Y component, placed in brackets.
<code>&lt;!-- %((Unit)) --&gt;</code>	The physical unit of a data object or its Y component, placed in parentheses.
<code>&lt;!-- %(UnitX) --&gt;</code>	The physical unit of a data object's X component.
<code>&lt;!-- %([UnitX]) --&gt;</code>	The physical unit of a data object's X component, placed in brackets.
<code>&lt;!-- %((UnitX)) --&gt;</code>	The physical unit of a data object's X component, placed in parentheses.
<code>&lt;!-- %(UnitZ) --&gt;</code>	The physical unit of a data object's Z component.
<code>&lt;!-- %([UnitZ]) --&gt;</code>	The physical unit of a data object's Z component, placed in brackets.

Field	Meaning
<!-- %((UnitZ)) -->	The physical unit of a data object's Z component, placed in parentheses.
<!-- %(User) -->	Name of the FlexPro user.

### Exporting a Complete Project Database

1. Select the root folder of the project database displayed in the first line of the Folders window.
2. Select **Export** from FlexPro's **File** menu.
3. Navigate to an empty folder where the web page is to be created.
4. Under **File Type** select **HTML Document** for the format.
5. Click on **Save** to start the HTML Export Wizard, which will guide you through the remaining steps.

### Exporting Individual Project Database Objects

1. Select all of the objects in the Object List that you want to export to the file.  
Alternatively, you can also select an object for export that is in a worksheet or document.
2. Select **Export** from FlexPro's **File** menu.
3. Navigate to an empty folder where the web page is to be created.
4. Under **File Type**, select **HTML Document** for the format.
5. Click on **Save** to start the HTML Export Wizard, which will guide you through the remaining steps.

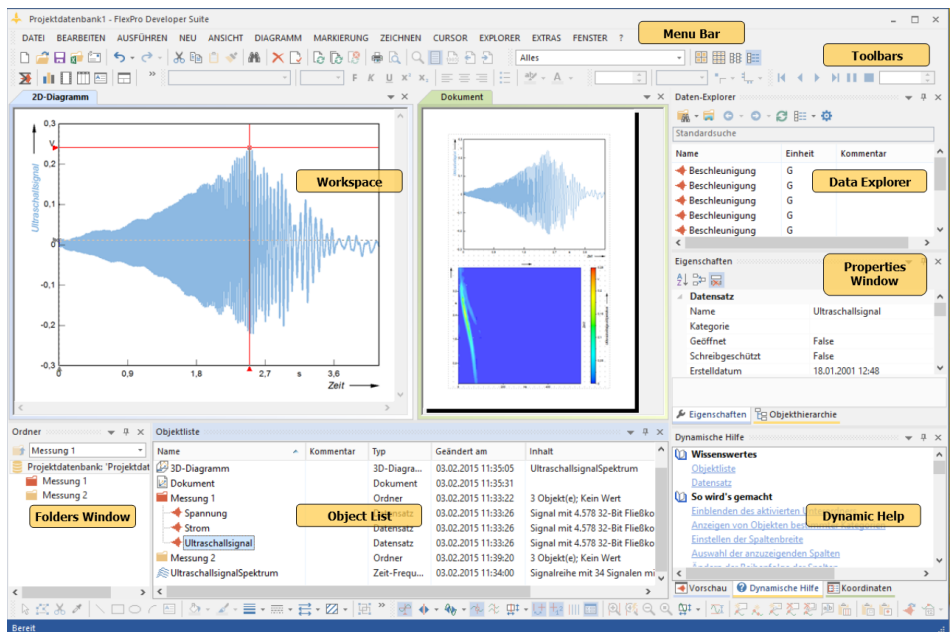
### Exporting Multiple Project Databases to a Single Web Document

1. Select the root folder of the first project database to be exported. The root folder is displayed in the first line of the Folders window.
2. Select **Export** from FlexPro's **File** menu.
3. Navigate to an empty folder where the web page is to be created.
4. Under **File Type**, select **HTML Document** for the format.
5. Click on **Save** to start the HTML Export Wizard, which will guide you through the

remaining steps.

6. Select the root folder of the next project database to be exported.
7. Select **Export** from FlexPro's **File** menu.
8. Click on **Save** without changing the folder beforehand.
9. You can now change settings in the HTML Export Wizard, if necessary. However, you should not change the settings for the table of contents.
10. Repeat steps 6 through 9 for each additional project database.

## 3.2 FlexPro Main Window



FlexPro's main window shows the data of an open project database and offers you options for navigating and searching within the project database as well as editing objects in the project database. The main menu consists of the following components:

### Menu Bar

This is displayed at the top of the main window. It offers all of the available commands

arranged in several menus to the open project database and, if applicable, to the currently opened object.

### **Toolbars**

The toolbars are displayed below the menu bar and at the bottom of the main window. They offer you quick access to the most important commands and options.

### **Folders**

In this window, which is usually displayed on the left-hand side of FlexPro's main window, you can view the entire folder hierarchy of your project database.

### **Object List**

All objects and subfolders of the folder selected in the Folders window are displayed in the Object List.

You can move, copy, rename and delete objects just as you would in Windows Explorer. What is different from Windows Explorer, however, is the fact that in Object List you can also re-sort data sets; for instance, you can stipulate a certain order when creating a diagram.

You can divide the objects in the Object List into categories to later be able to display only the objects from one or more categories.

Only the content of the folder selected in the Folders window is displayed in the Object List. You can display the content of additional folders by opening these in the workspace.

When you use your mouse to point to the name of an object in the Object List, a popup window appears for this object. If the object is a data object, all numerical properties of the object's data, such as extreme values or data structure, are displayed.

### **Object Hierarchy**

This window displays the objects of the folder selected in the Folders window hierarchically. You can therefore simply reconstruct which objects are linked directly or indirectly with a particular object. The display is clearer than the object list, since only the objects not used in any other object, such as your documents or worksheets, appear at the very top level. When you select an object in the hierarchy, the selection represents not only the object itself, but also all of the subordinate objects that it uses. This makes it easier for you to copy complete object networks, for instance.

### **Event Log**

All errors and warnings that occur when updating analyses and documents are displayed

in the Event Log with the time of their occurrence and the name of the source. Typical events include: data sets not found, syntax errors in formulas or calculation errors, such as a division by zero. The Event Log supports two modes: **Event History** and **Troubleshooting**. **Event History** logs all events chronologically, while **Troubleshooting** deletes old messages before each update so that only errors that still remain are displayed.

## **Search Results 1 & 2**

These windows show the objects found when searching in the project database.

## **Watch**

This window displays the result of the current statement as well as the contents of selected variables while you are debugging FScript code.

## **Preview**

The Preview window allows you to quickly view the content of a selected object.

## **Properties**

The powerful Properties window allows you to quickly edit the properties of selected objects.

## **ASAM ODS Data Source**

This window is only available when the ASAM ODS Data Import Option is installed. It displays the data from a linked ASAM ODS data source and lets you navigate and search within the tree structure as well as import measured data from the ASAM ODS data source to the FlexPro project database.

## **Data Explorer**

This window is only available when the Data Explorer Option is installed. The Data Explorer indexes the data on your hard drive. You can use it to quickly search through your data, view a preview with statistics and import data.

## **Dynamic Help**

This window displays selected sections and statements related to the object that you are currently processing or to the window in which you are currently working. Simply click on a help topic to display it.

## Workspace

This is the core area of the main window where FlexPro displays the open objects in the project database on tabs. You can divide the workspace horizontally and vertically into tab groups to display windows side by side.

## Context Menu

A context menu allows you to choose the most important commands for the current selection or the current working environment. You obtain a context menu for a certain object by clicking on the object with the right mouse button.

## Working with Windows

### Showing a Window

- In the **View** menu, point to **Window** and select a window you want to display.

### Closing a Window

- Click on **Close** to the right of the window title bar.

### Hiding an Anchored Window

- Select the option **Auto Hide** to the right of the window title bar.

---

#### Note:

- If the window is hidden automatically, only its title appears on the edge of the main window. If you use your mouse to point to this title, the window appears temporarily.
  - To display the window permanently, uncheck **Auto Hide**.
- 

## Moving a Window

### Freely Positioning an Anchored Window or Tab Group

1. Point to the title bar of the anchored window or tab group and press your left mouse button.
2. Drag the window to a location where no positioning arrow appears and release the mouse button.

### **Freely Positioning a Window in a Tab Group**

1. Point to the window's tab group and press the left mouse button.
2. Drag the window to a location where no positioning arrow appears and release the mouse button.

### **Anchoring a Floating Window to the Location Where It Was Last Anchored**

- Double-click on the window's title bar.

### **Anchoring a Window or Tab Group**

1. Point to the title bar of the anchored window or tab group and press your left mouse button.
2. Drag the window to a location where the positioning arrow appears. The resulting anchor position is now highlighted.
3. Release the mouse button.

### **Restoring the Window Arrangement**

- Select **Reset Window Arrangement** from the **Window** menu.

---

**Note:** All windows and toolbars in the main window that can be anchored are restored to their original arrangement.

---

### **Activating a Window**

1. Press and hold the CTRL key while pressing the TAB key to view a selection window.
2. Now press the TAB key multiple times until you see the window you want to view. Now release the CTRL key to open the selected window.

### **Working with the Object List**

#### **Selecting the Folder to be Displayed**

To display a folder in the Object List, click on the name of the desired folder in the Folders window. If the name of the folder to be displayed is not visible because it is located in a tree of the folder hierarchy that is not displayed, expand the corresponding tree by clicking on the '+' icons in front of the parent folders until the folder to be

displayed becomes visible.

If the folder immediately above the folder to be displayed is currently displayed in the Object List, double-click on the desired folder name in the Object List to display the folder.

**Selecting Objects**

You can use the known procedures from Windows Explorer in FlexPro to select objects. The following table offers a summary and presents two FlexPro-specific selection options:

What would you like to select? Procedure	
An object	Click on the desired object or move the selection onto the object using the arrow keys. This selection can also be made in the name row of the data view.
Multiple objects	Click on the desired objects while holding down the CTRL key. This selection can also be made in the name row of the data view.
Neighboring objects	Click on the first object to be selected. Press the SHIFT key and then click on the last object to be selected. This selection can also be made in the name row of the data view.
All	Use the <b>Select all</b> command from the Edit or context menu or use the CTRL+A shortcut.
Unselected objects	Use the <b>Invert Selection</b> command from the Edit or context menu.
Linked objects	Click on the desired object and run the <b>Select Linked Objects</b> command from the Selection or context menu. All objects in the folder displayed that the selected object requires for evaluation or display are also selected. This command makes it easier to move or copy complex analyses and presentation objects.

**Selecting the Display Type**

For the Object List, you can use the **View** menu, the context menu or the **Object List** toolbar to select one of four display options:

Command	Display
<b>Large Icons</b>	The objects in the folder to be displayed are arranged in rows,



Command	Display
	displaying the name of the object under a large icon that clarifies the object type.
<b>Small Icons</b>	Here, the objects are also arranged in rows, but a smaller icon is used, which is placed in front of the object name.
<b>List</b>	With this display option, the objects are arranged in columns, with a small icon and then the object names.
<b>Details</b>	The detail view uses a line for each object in the folder displayed. After the icon and the object name, the line contains the object comments, its file type in plain text, the creation date and a description of the contents of the object.

### Sorting the Object List

The list of objects displayed can be sorted in the Details view.

By clicking on one of the column headers at the top of the Object List, the rows of the list view are sorted in ascending order according to the criterion selected. The sorting column is represented by a triangle in the column header. If you then click again on the button of the current sorting column, this reverses the sorting order, which is then represented by an inverted triangle on the button.

In the name column there is a third mode in which no triangle is displayed. The objects are then displayed in the order defined by the user. In this mode you can sort the objects manually.

---

**Note:** Manual sort is not available if you display the objects in groups.

---

### Manually Sorting the Object List

To re-sort the objects displayed in the details view of the Object List, do the following:

1. Click on the **Name** column title in the Object List until no triangle appears. This chooses the user-defined order.
2. Select a number of objects that you want to move to a different place.
3. Use the mouse to drag the selected objects to the position where you want to insert them and drop them there. You can insert the objects between two objects, before the top object or under the bottom object. If you insert them between two objects, you have to make sure that both of the objects are highlighted so that you can drop in

the objects that you want to insert.

or

1. Click on the **Name** column title in the Object List until no triangle appears.
2. Select the objects you want to move.
3. Press the UP or DOWN key while holding down the ALT key to move to objects up or down one position.

---

### Note:

- When creating diagrams or tables, the data sets selected are assigned in order from the top to the bottom to the curves or columns respectively. If you place the data sets in the correct order from the beginning, you will avoid having to make corrections in the diagram or table later.
  - Manual sort is not available if you display the objects in groups.
- 

### Setting the Column Width

You can change the column width of the individual columns in the Object List's Details view by moving the mouse cursor in the column header area onto the dividing lines between the buttons. The mouse cursor then changes shape. The following actions are then possible:

- You can individually set the width of the column by using your left mouse button to click and then drag the line to the right or left.
- If you double-click, the width of the selected column is automatically adjusted so that all of the entries in this column can be read completely.

### Displaying Objects from Particular Categories

#### To display only the objects of a particular category in the Object List:

- Select the category from the **Choose Object Category** drop down list on the **Object List** toolbar.

#### To display objects from multiple categories in the Object List:

1. Select **Multiple Categories** from the **Choose Object Category** drop down list on the **Object List** toolbar.
2. In the dialog box that appears, select the categories to be displayed.

**To display objects from all categories in the Object List:**

- Select **All** from the **Choose Object Category** drop down list on the **Object List** toolbar.

**Displaying Several Folders**

In the Object List, only the content of the folder selected in the Folders window is displayed. You can display the content of additional folders by opening these in the workspace. To do this, double-click on a folder in the Folders window that you want to open, or select a folder from the Object List and choose **Open** from the context menu.

**Displaying the Parent Folder****To display the parent folder of the folder shown in the Object List:**

On the **Object List** toolbar, click on **Down** or press BACK.

**To display the parent folder of an object shown in a window:**

1. Right-click with your mouse on the object window tab.
2. In the context menu that appears, select **Display Parent Folder**.

**Activating and Deactivating a Subfolder****To activate a subfolder:**

- Select the folder from the **Activate or deactivate folder** list box at the top of the folder window.

**To deactivate a subfolder:**

- Select **(No activated subfolder)** from the **Activate or deactivate folder** list box.

**Displaying the Activated Subfolder**

You can display the content of the active subfolder in the Object List. To do this, select the option **Overlay Activated Folder** from the **View** menu. In the **Details**, the contents of the activated folder will be displayed indented below the folder. You can work normally with the displayed objects. For instance, you can select data sets to create

a diagram.

When accessing objects, objects from the activated subfolder override objects of the same name in the main folder. You can hide these overlapped objects by selecting the option **Hide Overlapped Objects** from the **View** menu. If you do not hide the objects, their icons will be displayed with less contrast in the Object List.

### Selecting the Columns to be Displayed

In the Object List's Details view, you can add or remove columns.

1. Right-click with your mouse on a column header.
2. In the context menu that appears, select a column to be displayed, or deselect a column that is to be removed.

---

**Note:** A newly added column appears to the right of the list. You can drag it with the mouse to another location.

---

### Changing the Column Order

In the Object List's Details view, you can change the column order.

- To do this, click on a column header and drag it to the desired location while holding down your mouse button.

### Displaying Objects in Groups

In the Details view, the list of objects displayed can be grouped according to the currently selected sorting column.

1. Right-click with your mouse in the Object List.
2. In the context menu select the option **Show in Groups**.

## Working with the Object Hierarchy

### Displaying Linked Objects

- In the object hierarchy, click on the plus symbol of the object with the linked objects that you want to view.
- You can view the objects linked indirectly with the object by opening additional levels the same way.

**Displaying the Purpose of Linked Objects**

1. In the object hierarchy click on the plus symbol of an object to view its linked objects.
2. Now point with your mouse on a linked object to view the popup window showing the purpose of the object.

**Selecting an Object and Linked Objects**

- Click on the name of the object in the object hierarchy to select it.

---

**Note:** You can select only one object at a time in the object hierarchy. However, by selecting the object, all objects dependent on the selected object are selected as well. This makes it possible to copy or move a selected object while simultaneously copying or moving all dependent objects with it.

---

**Working with the Properties Window****Editing Properties**

1. Select all objects for which you want to edit a particular property, for instance, the labels for the X and Y axes of a diagram or several data sets and formulas in the Object List. You can select several objects in presentation objects by holding down the CTRL key and clicking on them. With the ALT key pressed, you can select all elements of the same type, such as the symbols on all curves.
2. Click on the edit box in the Properties window for the property you want to edit. If you are working with a numerical value or text, you can enter the value or text directly. If you are working with a drop down list box, then you can open a list and make a selection. Otherwise, a button is displayed to the right of the field so that you can open a dialog box for editing.
3. You can finish your input using the ENTER key. The change will affect all selected objects containing this property.

---

**Note:**

- On the bottom of the Properties window, a description for the selected property is displayed.
  - If the edit box for the selected property appears empty, then you are either working with an empty text field or the property occurs in several objects with different content. In any case, your input is entered into all selected objects containing the edited property.
-

### Showing or Hiding Properties Not Available in All Objects

If you have selected several different objects, e.g. data sets and formulas in the Object List or curves or axes in a diagram, you can show or hide properties that are not common to all objects:

- Select the option **Show Only Common Properties** from the Properties window toolbar to hide these properties, or uncheck the option to show them.

---

**Note:** Properties that occur in several objects and that have different content are displayed with an empty edit box. If you edit this type of property, then this change will affect all selected objects.

---

### Showing or Hiding a Properties Group

- To show or hide a properties group, click on the plus or minus icon in the Properties window to the left of the grouping icon.

### Showing or Hiding Properties with Read-Only Access

Properties with read-only access appear in gray. You can show or hide these types of properties:

- Select the option **Show Only Changeable Properties** from the Properties window toolbar to hide these properties, or uncheck the option to show them.

### Sorting or Displaying in Groups of Properties

You can display the properties shown in the Properties window either as a sorted list or in groups.

- Select the option **Sort Properties** from the Properties window toolbar to display the properties as a sorted list, or uncheck the option to show them in one or more named groups.

---

**Note:** The groups correspond to the sub-objects of the selected object, e.g. the connection line or symbols of a selected curve.

---

## Working with the Event Log

### Fixing the Cause of Events

- Double-click on the event for which you want to fix the cause. The object that triggered the event opens, and you can fix the error.

### Clearing the Event Log

1. Right-click with your mouse in the Event Log to open the context menu.
2. Click on **Clear**.

### Customizing Event Handling

You can customize event handling in FlexPro to meet your specific requirements. For example, you can specify a maximum number of events that should be collected in the log and whether the log should be saved in the project database.

1. Right-click with your mouse in the Event Log to open the context menu.
2. Click on **Properties**.

## 3.3 FlexPro Objects

All data, analyses and presentations are stored by FlexPro as objects in the project database. These objects are linked to one another and thus form a dynamic network. For instance, a document is linked to the diagram within it, the diagram is linked to an analysis object and the analysis object is linked to the data set containing the raw data.

### Updating

After editing one or more objects, they will have to be updated. In certain instances, updating can require a substantial number of calculations, which is why there are various options available to you to manage the update process. You can manually update individually selected objects or all open objects. FlexPro then analyzes the object network and updates only those elements that are not current. You can, however, force a complete update. If you enable automatic update, then FlexPro updates the objects automatically when they are opened or when changes are made to them. FlexPro can update objects in the background. This allows you to continue working during the update. Objects that are currently being updated, however, are blocked from use.

If you work with data links that refer to changing data, you can enable periodic updating. FlexPro will then refresh the display automatically at the specified rate.

### Read-Only and Lock Attributes

You can set every object in the FlexPro project database to read-only. You will no longer be able to edit the attribute of a read-only object, but you can continue updating the object, e.g. after you import new raw data. If you use cursors in a read-only diagram, all functions that would normally change the diagram, such as zooming, setting and removing markers, or translating curves, are blocked. Even folders can be set to read-only. In a read-only folder you cannot add or remove objects. You can, however, continue to edit the objects themselves as long as they are not also read-only.

You can lock the object to provide further protection for an object. A locked object can no longer be updated. For instance, a locked diagram behaves like a static image, and a locked formula, analysis object or data link is like a data set. The data of locked formulas, analysis objects and data links are stored in the project database and remain unchanged after the database is reopened. You cannot use the cursors in a locked diagram or document. Worksheets cannot be locked.

### Properties Dialog Box and Window

A Properties dialog box is assigned to each object. Here, you can adjust all of the object-specific settings. For example, you can change the comments for a data set or the options of an analysis object.

There are often more settings options available in the Properties dialog box than in the editor that you normally use to edit the object. Thus, for example, you can also set the order in which the curves are to be drawn in the Properties dialog box of a diagram. Often there is no access in the Editor to the item you would like to edit, e.g. if you would like to delete a curve from a diagram containing a data set that does not provide any data. The curve is then no longer visible in the Editor and can therefore not be clicked. You also have to use the Properties dialog box in such cases.

FlexPro also has a Properties window in which all properties of the selected objects are displayed as a list. The advantage of the Properties window over the Properties dialog box is that you can edit the properties of several objects at once. For instance, to enter the physical unit "V" in ten data sets simultaneously, select these in the Object List and enter "V" in the **Y Unit** field.

### Object Parameters

In every FlexPro object, including folders, FlexPro manages a list of parameters that you can edit freely. Each parameter can take a scalar value in one of the data types supported by FlexPro. You can also assign a physical unit to the value. You can later query the parameters using a formula and calculate or output them. Some import filters store additional information as parameters.



## Naming Conventions for Objects

The following points must be taken into consideration when assigning names to FlexPro objects:

- The maximum length of an object name is 200 characters.
- The following characters must not be used: \ / : . \* ? ! " ' < > |
- No differentiation is made between upper and lower case when objects are saved, i.e. the names "OBJECT" and "Object" are considered identical.
- Objects of different types can have identical names, where data sets and formulas are considered one object type in this case.

---

**Note:** A name should start with an alpha character and only alpha characters and numerical digits should be used for the rest of the name. This corresponds to the syntax rule for identifiers in FPScript.

---

## File Name Extension

In a project database folder, all objects of the same type have to have a unique name. Thus, it is not possible, for instance, to place two documents with the name "Document" in the same folder. However, you can use the same name for objects of different types, e.g. a diagram and a column table both with the name "Evaluation". This means that the name of an object is not enough to clearly identify an object.

Therefore, a file name extension is assigned to each object class. This extension is appended to the name to clearly identify it. The following table shows the file name extensions used:

Object Class	File Name Extension
Data Set, Formula, Analysis Object, Data Link Object, etc.	(no name extension)
Document	.doc
Worksheet	.wks
2D Diagram	.2d
3D Diagram	.3d
Column Table	.tab
Cell Table	.ttb
Text	.txt
Folder, Excel Workbook	.fld

As you can see from the table, you do not have to specify a file name extension for data sets, formulas and objects derived from them. This is practical because these are the most frequently accessed objects.

To access the "Data" data set comments within a formula, write:

`Data.Comments`

If you would like to access the comments from a document called "Evaluation", you have to use the following expression:

`Evaluation.doc.Comments`

## Working with Objects

### Creating Objects

1. In the Folders window, select the folder where you want to insert the object to display this folder in the Object List.
2. Now select all of the data to be displayed in the object or to be calculated by the object. If you want to select data, you can highlight entire data sets within the Object List or within the name row of the open folder's data view, or you can select a cell range in a data set window or in the data view.
3. Select the type of the newly created object from the **New** menu or using the corresponding icon on the **FlexPro Objects** toolbar.
4. An object of the selected type is set up with a unique name in the folder displayed, and, if necessary, a wizard is first displayed to help you when setting the object properties.

---

**Note:** You can also embed diagrams, tables and text objects to documents or worksheets. These will not appear in the Object List.

---

### Updating Objects

#### To update one or more objects:

1. Select the objects in the Object List or an object in the hierarchy view.
2. Click on **Update** in the **Standard** toolbar.

**To update all objects in a folder:**

1. Select the folder in the Folders window.
2. Click on **Update** in the **Standard** toolbar.

---

**Note:** Objects contained in subfolders are only updated if they are used by objects in the selected folder.

---

**To update an object that is open:**

1. If necessary, click on the tab of the window where the object is displayed to activate it.
2. Click on **Update** in the **Standard** toolbar.

**To update all objects that are open:**

- Click on **Update All** in the **Standard** toolbar.

---

**Note:** If instead of using the **Update** menu, select **Force Update** command, the selected objects will be fully updated even if FlexPro was unable to detect any relevant changes.

---

**To update the active object or all opened objects periodically:**

1. Choose the **Set Cycle Duration** command from the **Run** menu.
2. Enter the preferred cycle duration using the format HH:MM:SS.
3. If the command **Update All** is to be performed in cycles, select the option **Update All**. Otherwise, FlexPro uses the **Update** command.
4. Click on OK to close the dialog box.
5. Select the option **Update Periodically** from the **Run** menu. The cyclical update will take place until you uncheck the option again.

**Updating Objects in the Background**

1. Select **Options** from the **Tools** menu.
2. Click on the **System Settings** tab.
3. Select the option **Update objects in the background**.

---

**Note:** The option also affects updating of Preview.

---

### Changing the Update Interval for Periodic Update

1. Choose the **Change Update Interval** command from the **Run** menu.
2. In the dialog box you can set the interval of time for how frequently you want the **Update** command to run on a reoccurring basis.
3. Choose whether the **Update** or **Update All** command is to be used.

### Changing Lock and Read-Only Object Attributes

#### To change the lock or read-only attribute of one or more objects:

1. Select the objects in the Object List.
2. Right-click with your mouse on the selection to check or uncheck the **Locked** or **Read-only** option in the context menu.

#### To change the lock or read-only attribute of an object and all linked objects:

- Right-click with your mouse on the object in the hierarchy view and check or uncheck the **Locked** or **Read-only** option in the context menu.

#### To change the lock or read-only attribute of all objects in a folder:

1. Right-click with your mouse on the folder in the Folders window and check or uncheck the **Locked** or **Read-only** option in the context menu.
2. In the dialog box that appears, specify whether the attribute is to be changed for the folder alone or for all objects and subfolders.

### Categorizing Objects

1. In the Object List, select the objects you would like to assign to a new or existing category.
2. Choose the **Categorize** command from the context menu or the **Explorer** menu.
3. In the dialog box displayed, enter the desired category or select one of the existing

categories from the list box.

### **Opening an Object**

#### **To open a folder or Excel workbook in a separate window:**

- Double-click on the name of the folder or Excel workbook in the Folders window or in the object hierarchy.

#### **To open the formula of an analysis object or data link object:**

- Right-click with your mouse on the name of the object in the Object List or in the object hierarchy and select **Open** from the context menu.

#### **To open any other object:**

- Double-click on the name of the object in the Object List or object hierarchy.

### **Undoing and Restoring Edits Made to an Object**

#### **a) In the Object List:**

1. Right-click with your mouse on the object.
2. In the object context menu that appears, select **Undo** or **Restore**.

#### **b) In the open window of the object:**

1. Select **Undo** or **Restore** from the **Edit** menu.

### **Displaying an Object's Properties Dialog Box**

- Right-click with your mouse on the object in the Object List, object hierarchy or Folders window and choose **Properties** from the context menu.

### **Renaming an Object**

1. Select the object to be renamed from the Object List, object hierarchy or data view of an open folder.
2. Press the F2 key or click on the object that is already highlighted. An input field appears, in which the object name is selected.

3. Enter a new name or change the existing name.

## Moving Objects

### To move objects via the Clipboard:

1. Select the objects in the Object List that you want to move.
2. From the **Edit** menu, select **Cut**.
3. Open the folder where you would like to move the object. You can also open the target folder in another project database.
4. In the **Edit** menu, click **Paste**.

### To move objects with the mouse:

1. Select the objects in the Object List that you want to move.
2. Use your mouse to click on one of the objects selected. While holding down the left mouse button, move the mouse cursor onto the target folder and once there, release the mouse button.

---

**Note:** If you move an object with a name that has already been used in the target folder, then FlexPro's behavior depends on which settings you chose on the **Operation** tab of the **Options** dialog box. If you enabled **Automatic renaming of objects with copying or moving actions**, the object names affected are made unique by appending a number to them in the target folder. Otherwise, a dialog box appears, allowing you to decide whether you would like to cancel the copying process, rename the object or replace the object in the target folder.

---

## Copying Objects

### To copy objects via the Clipboard:

1. In the Object List, select the objects you would like to copy.
2. From the **Edit** menu, select the command **Copy**.
3. Open the folder where you would like to save the copy. You can also open the target folder in another project database.
4. In the **Edit** menu, click on **Paste**.

**To copy objects using the mouse:**

1. In the Object List, select the objects you would like to copy.
2. Hold down the CTRL key and use your mouse to click on one of the selected objects. While holding down the left mouse button, move the mouse cursor onto the target folder and once there, release the mouse button.

---

**Note:** If you copy an object with a name that is already in use in the target folder, the behavior of FlexPro will depend on which setting you selected on the **Operation** tab in the **Options** dialog box. If you enabled **Automatic renaming of objects with copying or moving actions**, the object names affected are made unique by appending a number to them in the target folder. Otherwise, a dialog box appears, allowing you to decide whether you would like to cancel the copying process, rename the object or replace the object in the target folder.

---

**Copying Objects and Linked Objects**

To copy FlexPro objects with linked objects, you first have to select them in the Object List or object hierarchy. To do this, proceed as follows:

**In the Object List:**

1. In the Object List, select the objects you would like to copy.
2. Click with your right mouse button on one of the selected objects and choose **Select Linked Objects** from the context menu.

**In the object hierarchy:**

- In the object hierarchy, select the object you would like to copy.

---

**Note:** You can select only one object at a time in the object hierarchy. This selection, however, does not represent only the object itself, but also represents all objects under it in the hierarchy.

---

**To copy the selected objects via the Clipboard:**

1. From the **Edit** menu, select **Copy**.
2. Open the folder where you want to place the copy by selecting it in the Folders window. You can also open the target folder in another project database.

3. In the **Edit** menu, click **Paste**.

**To copy the selected objects with the mouse:**

4. Hold down the CTRL key and use your mouse to click on one of the selected objects. While holding down the left mouse button, move the mouse cursor onto the target folder and once there, release the mouse button.

---

**Note:** If the objects to be copied need to be renamed in the target location because that location already contains objects with the same name, then FlexPro corrects all links so that the copied object network continues to work properly in the target location and no links are broken.

---

**Deleting Objects**

1. From the Object List, select the objects you want to delete.
2. From the **Edit** menu, select **Delete** or press the DEL key.

---

**Note:** You can also delete a single object or folder in the object hierarchy or in the Folders window.

---

**Editing Object Parameters**

**In the Properties window:**

In the Properties window you can edit the parameters of one or more objects at the same time.

1. Select one or more objects with parameters that you would like to change.
2. Scroll down the Properties window until you see the **Parameters** group.
3. Enter the new values for the parameters that you want to change.

**In the Properties dialog box:**

In the Properties dialog box you can edit parameters as well as add, remove or resort them.

1. Right-click with your mouse on the object in the Object List and select **Properties**.
2. Switch to the tab called **Parameters**.



**To add a parameter:**

3. Click **Add Parameter** or press ENTER.
4. In the **Name** field, enter a unique **name** for the parameter.
5. Choose the desired data type in the **Data Type** field.
6. In the **Value** field, enter the value of the parameter and append the unit symbol, if applicable.

**To change a parameter:**

3. Select the parameter in the **Parameters** list.
4. Change the name, data type, value or unit and exit the input field with the TAB key for the change to take effect.

**To delete a parameter:**

3. Select the parameter in the **Parameters** list.
4. Click **Remove Parameter** or press the DEL key.

**To sort the parameters in the list:**

3. Select a parameter in the **Parameters** list.
4. Click **Move Parameter Up** to move it up one position.
5. Click **Move Parameter Down** to move it down one position.

---

**Note:** You can use FPScript formulas to enter data. To do this, the text to be entered must start with '=' for instance, the value 5 is entered for the parameter when you input '=2 + 3'.

---

## Printing Objects

**To print several FlexPro objects:**

1. From the Object List, select the objects you want to print.
2. From the **File** menu, select the command **Print**.
3. In the **Print Setup** dialog box that appears, select the printer and the page layout.
4. Click on OK to start printing.

---

**Note:** To cancel printing, press the ESC key.

---

**To print an object displayed in a window:**

1. Click on the object in the window that you want to print.
2. From the **File** menu, select the command **Print**.
3. In the **Print** dialog box that appears, select the printer, the print range and the number of copies.
4. Click on OK to start printing.

### 3.4 Data Objects

Data object is the term for data sets, formulas, analysis objects and data link objects, or for any objects that contain data or return data as a calculated result. All data objects except data sets are based on FPScript code. These data objects, however, can be used like data sets and can be displayed, for instance, as a diagram or table. The result can have one of the data structures and data types supported by FPScript. You can convert all FPScript-based data objects into data sets at any time. The formula containing the FPScript is calculated one last time and then is replaced by its result.

#### Data Type

FlexPro is capable of managing and processing the data so that the memory required is optimally adapted to the resolution of the data. This is particularly useful for large data sets, in view of the fact that the memory required can vary by up to a factor of four, depending on the data type.

FlexPro achieves this by using different data types, i. e. storage formats for numbers and text. The system has over four numerical basic data types that can be used for real and complex numbers as well as numeric data types for calendar time and time span. In addition to this, there is a data type for strings, for Boolean values and for the **Empty** data type, which represents an empty result.

The following table provides an overview of the data types.

Data Type	Explanation	Notation in FPScript
Empty	Uninitialized variable, empty result or placeholder for a function argument.	Empty <Identifier>
Boolean value	Can have the value TRUE or FALSE.	TRUE, FALSE
16-bit integer	Can contain integral numbers in the range from -32768 to +32767.	1234S, -1245S
32-bit integer	Can contain integral numbers in the range from -2147483648 to +2147483647.	1234N, -1234N, 1000000N

<b>Data Type</b>	<b>Explanation</b>	<b>Notation in FPScript</b>
64-bit integer	Can contain integral numbers in the range from -9223372036854775808 to +9223372036854775807.	1234L, -1234L, 2147483648N
32-bit floating point value	Can contain floating point values in the range from -3.4e+38 to +3.4e+38.	1.23S, 1.45e-3S, ?S
64-bit floating point value	Can contain floating point values in the range from -1.7e+308 to +1.7e+308.	1.23, 1.45e-3, 1.23L, ?
Complex 16-bit integer	Can contain complex integers in the range from -32768 to +32767.	(1234S, -1245S)
Complex 32-bit integer	Can contain complex integers in the range from -2147483648 to +2147483647.	(1234N, -1234N)
Complex 64-bit integer	Can contain complex integers in the range from -9223372036854775808 to +9223372036854775807.	(1234L, -1234L)
Complex 32-bit floating point value	Can contain complex floating point values in the range from -3.4e+38 to +3.4e+38.	(1.23S, 1.45e-3S)
Complex 64-bit floating point value	Can contain complex floating point values in the range from -1.7e+308 to +1.7e+308.	(1.23, 1.45e-3)
Calendar Time	Can contain data and time values. Calendar time values are stored as 64-bit floating point values representing the number of seconds that have passed since 1/1/1970 Coordinated Universal Time (UTC). Calendar time values before this date are not supported, i.e. the floating point value based on this calendar time value cannot be negative. FlexPro interprets calendar time values as quantities using seconds as the implicit unit. The unit symbol, however, is not output.	'23.12.2004 12:23:30.1234' '12/23/2004 12:23:30.1234' '?/?/?' (Elements of time can be omitted starting from the right and are then taken as 0)
Time Span	Can contain time spans written in the formats DD:HH:MM:SS or HH:MM:SS. The values are stored as 64-bit floating point values with the unit as seconds. FlexPro interprets time spans as quantities using seconds as the implicit unit. The unit symbol, however, is not output.	'12:23:30.1234' '11:12:23:30.1234' '-12:23:30' (negative time span) '??:?:?'
String	Can contain strings of up to 2 GB in length.	"This is a string"
Object Reference	A reference to any automation object or	Set Object = Diagram.2d

Data Type	Explanation	Notation in FPScript
	FlexPro object.	

As can be seen from the examples above, FlexPro uses the decimal point or the appended 'N' or 'n' to distinguish between integers and floating point values. The resolution can be specified using the appended 'S' or 'L', or 's' or 'l'. 'S' stands for "Short", i.e. a 16-bit integer or 32-bit floating point value, and 'L' stands for "Long", i.e. a 64-bit integer or 64-bit floating point value. If you omit the suffix, the higher resolution is automatically selected for floating point values, and for integers, a 32-bit or 64-bit resolution is used, depending on which format can store the specified constant. The 'L' can therefore usually be omitted.

**Note:** How decimal numbers to which you have not added a suffix or decimal point are interpreted depends on the setting **Interpret decimal numbers without a decimal point as floating point values** on the **General** tab of the database Properties dialog box. This option is enabled by default, which means that the numbers are interpreted as 64-bit floating point values. This setting prevents accidental rounding errors that may occur when using integer arithmetic. Integral division  $1 / 2$  results in 0 and not 0.5, for instance.

If you are calculating data with a different format, FlexPro adapts the data types automatically. Automatic adaptation always happens in such a way that no information is lost. If, for instance, an integer and a floating point value are to be added together, the integer is converted to a floating point value and not the other way round, because otherwise decimal places might be lost. For constant data, which you use in formulas, FlexPro recognizes the data type from the notation. You have the option of converting the data types explicitly by using an operator for type conversion :

`FloatingPoint64` `DataSetSeries` converts all values in data series into 64-bit floating point values.

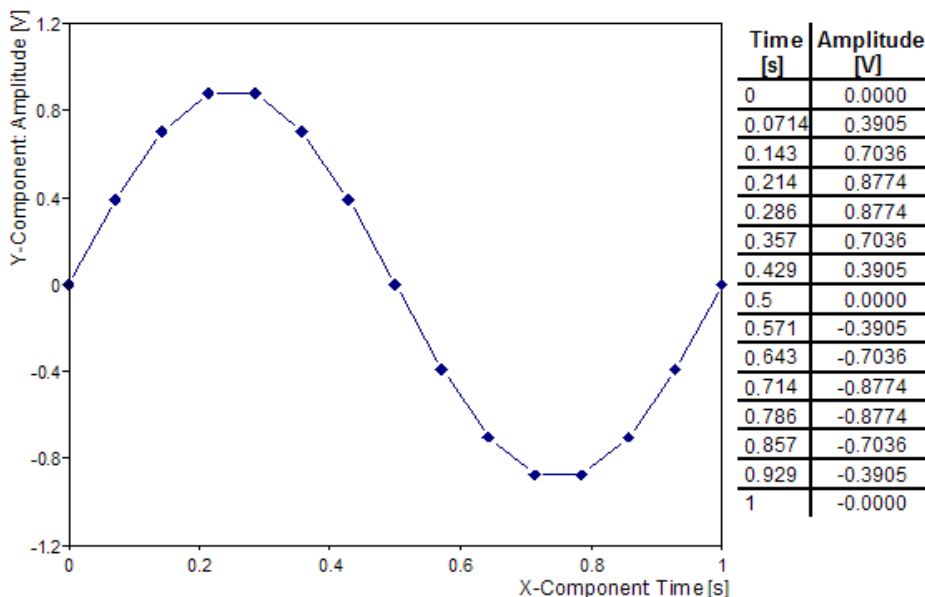
**Object references** can only be used on a limited basis. You can assign an object reference to a local variable but neither pass it as an argument to an FPScript function nor return it as the result of a formula. If an object reference is used as a function argument, as a return value of a formula, or in a mathematical calculation, then it is converted into its value. If the object, to which the reference refers, is a data object, e.g. a formula, then its return value is used; for all other objects, the value is Empty.

Component

Data sets can have an aggregate data structure, i.e. a data structure that consists of up to three components. These components are called the **X, Y and Z components**. The description corresponds to the names of the axes of 2D and 3D diagrams.

A simple time signal consists, for example, of a Y component, which contains a data series with the voltages, and an X component, which contains a data series of equal length with the associated times. One value each from the Y and X component together form one point of a curve in the two-dimensional plane of a 2D diagram.

The following illustration clarifies these relationships:



Within formulas, you can separately access the individual components of the data set by specifying the name of the component as a name extension. `Signal.Y` extracts, for instance, the Y component from the data set `Signal`.

## Data Structure

Data sets can have different data structures. The data structure specifies how the data is organized in the data set. A differentiation is made between simple data structures and aggregate data structures. Aggregate data structures are built up from the simple data structures. The following table provides an overview of **simple data structures**:

Data Structure	Contents
Scalar Value	A scalar value.

Data Structure	Contents
Data Series	A one-dimensional array of values.
Data Matrix	A two-dimensional array of values. The columns can be regarded as data series with the same number of values.

All data types are permitted for the simple data structures.

The following **aggregate data structures** are based on the structures above:

Data Structure	Contents	Example of Application
Signal	A Y component and an X component, each with a data set of the same length.	A time signal. The Y component contains the voltage and the X component the associated time.
Signal Series	A Y component with a data matrix and an X component with a data series. The length of the data series must match the number of rows in the data matrix.	A spectral series. The Y component contains several frequency spectra and the X component of the frequency values applicable for all spectra.
Signal series with a two-dimensional X component	A Y component and an X component with a data matrix of the same size.	A series of time signals. The Y component contains the voltages of the individual signals and the X component, contains the associated calendar times.
Signal series with a Z component	As with a signal series, however, with an additional Z component with a number of values that must correspond to the number of columns in the Y data matrix.	A spectral series, where the time points for which the spectra have been determined are entered into the Z component.
Signal series with two-dimensional X component and Z component	As with a signal series with a two-dimensional X component, however, with a Z component with a number of values that must correspond to the number of columns in the Y data matrix.	A series of time signals. The Y component contains the voltages of the individual signals and the X component contains the associated times in seconds. The Z component contains the calendar times at which the signals were measured.
Space Curve	A Y component, an X component and a Z component, each with a data series of the same length.	A trajectory in three-dimensional space. A measurement that was taken at arbitrary locations on a surface.
List	A list in which each element can have any data structure and can especially be a list itself. The	A list containing several data sets, such as the result of a data query. A complex data structure, such as

Data Structure	Contents	Example of Application
	elements in lists can also contain object references.	the result of a regression, which, in addition to a data series with found parameters, also contains their confidence intervals and the goodness-of-fit measure of the regression.

For the X and Z components of the aggregate data structures, all real data types as well as the calendar time and time span types are allowed. For the Y component, additional complex data types and the Boolean value type are permitted. The aggregate data structures from FlexPro especially simplify the analysis in FlexPro, since the X, Y and, if applicable, Z components in a single formula can be calculated. A formula that calculates a Fourier transform, for instance, can re-calculate the amplitudes of the time signal in spectral amplitudes and the time values into frequency values at the same time.

**Data Structure List** Use the list to create complex data structures, for instance, if the data structures supported by FlexPro are not suitable for the result. You can have the results returned as a list even when a calculation procedure returns several results that cannot be calculated independently of each other as separate formulas. Lists are the basis for multichannel analyses using FlexPro. Many analysis objects as well as functions and operators in FPScript support lists. They are processed by element and the result is then also a list.

Lists are special because they are only allowed as a return value for formulas, but not as a data structure for a data set. When converting a formula to a data set, only the first element in the list is used. The same applies to FPScript functions that do not support processing of lists. These also usually extract the first element and ignore the rest.

## Data Unit

Each component of a data object can have a physical unit and is thus converted to a **quantity**. The Unit Manager is an integral part of FPScript. You can make calculations with quantities directly in FPScript. FPScript processes the unit as an integral part of the data, is able to convert units, recognizes incorrect calculations and determines the result unit of a calculation from the units of the arguments. The following FPScript code generates, for instance, a signal with a sine wave with an amplitude of 2 V, a frequency of 0.1 kHz and a sampling interval of 1 ms:

```
// Increasing time values in the interval [0 s, 10 s]
// and sampling interval 1 ms
Dim t = Series(0 s, 10 s, 1 ms)
Signal(2 V * Sin(2 Pi * 0.1 kHz * t), t)
```

Alternatively, the unit can also be assigned subsequently via the Header Information of the data object. Please see below for further details.

### Assigning Data Objects

FlexPro normally saves all components of an aggregate data structure together in a single data set; for instance, the amplitudes and time values of a time signal. This also applies if there are several data sets with the same X component, such as when you import data from a multi-channel measurement. The time values for the individual channels are usually linearly increasing, so FlexPro only has to store the initial value, increment and number of values as the parameters instead of columns of numbers. There are, however, cases where the X values must be stored explicitly, e.g., because the sampling rate varied during the measuring process. In this case, storing the same X values in each data set would be inefficient. Therefore, FlexPro supports another option to manage aggregate data structures. To do this, the Y values and the X values are stored in separate data sets as data series, and these data sets are then assigned to each other.

This assignment is stored in the header information of the data set and can be changed on the **Component & Assignments** tab of the Properties dialog box.

The data set that contains the data series with the X values is marked as the X component there, and the data sets containing the Y values for individual channels of a measurement are marked as Y components respectively. In addition, reference to the X data set is made in the header information of the Y data set.

By making assignments, FlexPro can always find the appropriate X data set, e.g., when creating a diagram. In the diagram's curve, FlexPro then activates the option **Separate data sets for X and Y components** and additionally enters the X data set assigned in the Y data set into the curve.

Not only can this assignment be carried out for data sets, but also for all data objects, such as formulas. For formulas, however, the assignment is usually not necessary, since the X component can simply be included in the result with the Signal function.

### Data Order

The data order is a property of a data series, which specifies how the values in the data series are sorted. A distinction can be made between the following data orders:

Description	Purpose
None	The values in the data set are not in any particular order.
Constant	The data set is a scalar value, or all values in the data set are identical.



Description	Purpose
Increasing	Each value in the data set is greater than or equal to its predecessor.
Linearly increasing	The distance between neighboring values is constant and greater than zero.
Decreasing	Each value in the data set is less than or equal to its predecessor.
Linearly decreasing	The distance between neighboring values is constant and less than zero.

A signal for which the X component is linearly increasing is described as sampled equidistantly. Some algorithms for data analysis, e.g., FFT, require this.

FlexPro displays the data order of the individual data object components in a popup window when you use the mouse to point to the object in the Object List.

## Header Information

Header information, also known as metadata or data header, are attributes that you can assign to a data object.

For this purpose, you can enter the following in the Properties dialog box of a data object (the first four attributes apply to each object, not only for data objects):

- **Comments**

These are comments concerning the content of the object. The comments appear together with the name in the **Object List**. You can enter up to three comments for data objects with an aggregate data structure. One for each component.

- **Hyperlink**

A reference to an external file, website, e-mail address or another object in the project database. The **Follow Hyperlink** command opens the object to which it referred.

- **Date**

The object's creation date and time are specified here.

- **Parameters**

A freely editable list of parameters in which additional information can be stored.

The following attributes are specific to data objects:

- **Unit**

The physical unit for the individual components of the data object. The interpretation of the unit in the header information depends on whether the data object is a data set or a formula, or an analysis object:

- The unit is assigned to the data of a data set without transformation, and in the future these will be returned as present in this unit. In data sets, FlexPro generally always keeps the data unit in sync with the unit in the header information.
- The data of a formula or analysis object is, however, transformed to this unit before each access. If FPScript code provides a result without a unit, the specified unit will be assigned without transformation. However, if the code provides a result with a unit, this will be transformed into the specified unit. If the unit in the header information is incompatible with the unit that the result of the FPScript formula provides, then you will see an error message from FlexPro.

---

**Note:** To transform the data of a data set into a different unit, use the command **Change Units** or enter the new unit symbol in the **Data Unit** field of the Properties window.

---

- **Quantity**

The name of the physical quantity for the individual components of the data object.

- **Upper and lower data range limits**

With measured data, these fields specify the data range used in the physical unit of the data, such as -5 V to +5 V. FlexPro can use them for autoscaling axes. The data range limits are optional. If you do not specify them, the minimum and maximum in the data set will be used instead.

- **Time Stamp**

FlexPro uses these fields, if specified, to convert time data from relative time to absolute time and vice versa. The timestamp is a calendar time value, which is assigned the time  $t = 0$  in the time data. When converting to absolute time, this is added to all data, and when converting to relative time, it is subtracted.

- **Author**

The author of the object. This is particularly useful for data sets entered manually and for formulas.

- **Origin**

A description of the origin of the data object, e.g. the name of the location of a measurement.

- **Assigned data sets**

Refers to assigned data objects that form an aggregate data structure together with the object's data (see above).

## Data Export

FlexPro offers you the option of exporting data sets, formulas and related objects. You can save the data in the **FlexPro** (.fpf), **Text** (.txt), **CSV** (.csv), **Audio file** (.wav) file formats as well as several proprietary file formats. **CSV** stands for **Comma Separated Values**. The values are separated in this format with the list separator specified in the Windows **Control Panel** under **Region and Language Options**.

When saving in CSV or text format, you have the option of exporting several objects below one other or side-by-side to a single file. For these formats, you can additionally select different file formats and specify whether the result or formula code is to be output for the formulas.

The FlexPro format is similar to that variant of the text format, where the data of several data sets are stored side-by-side in a table and with all of the header information. Here, however, an additional row with a data format identifier is then output. Regardless of the system settings, floating point values are always output with a point as the decimal symbol. Calendar times are also always output in the same format, regardless of the system settings.

Primarily signals and signal series with two signals are suitable for export as an audio file. Signals are treated as a mono signal, and signal series with two signals are treated as a stereo signal. Signal series with more than two channels are exported as multi-channel wave files accordingly. The data's X component must correspond to one of the following sampling rates supported by the audio file format: 11025 Hz, 22050 Hz or 44100 Hz.

You can export the data from formulas, data sets and related objects as OLE objects and thus insert links to this data into OLE client applications. FlexPro provides the data in CSV and text formats.

## Working with Data Objects

### Playing Data Objects as an Audio File

1. In the Object List, use the right mouse button to click on a data set, a formula, an analysis object or a data link object.
2. In the context menu that now appears, choose the **Play** command.

---

**Note:** The data's X component must correspond to one of the following sampling rates supported by the audio file format: 11025 Hz, 22050 Hz or 44100 Hz.

---

### Editing a Data Object's Header Information

1. Use your right mouse button to click in the Object List on the formula or in the Formula Editor or data set window to open the context menu.
2. Select **Properties** to open the Properties Dialog Box.
3. You can make changes to all settings on the **General** tab.

---

**Note:** Analysis objects can often automatically generate the units and comments for your result. You can activate this mechanism by leaving the corresponding text boxes empty.

---

### Changing Data Object Units

#### Converting Units of Data Sets and Formulas

Only components that already have a unit are transformed to the new unit. Otherwise, the unit is only assigned without changing the data.

##### a) Via the Properties window

1. Select the data sets that have a unit you want to convert. Make sure to select only data objects with the same unit.
2. In the Properties window, enter the new unit in the fields **Y**, **X** or **Z Data Unit**.

##### b) Via the Replace or Convert Units dialog box

1. In the Object List, select the formulas and data sets that have a unit you want to convert. You can also select data objects that have different units.
2. Right-click with your mouse on the selected objects and choose **Change Units** from the context menu.
3. In the **Replace or Convert Units** dialog box enter the new units for the units that you want to convert.
4. For the **Action** select **convert**.
5. End the process by clicking **OK**.

#### Replacing Units of Data Sets

In this process, the units are replaced without changing the data.

##### a) Via the Properties window

1. Select the data sets that have a unit you want to replace.

2. In the Properties window, enter the new unit in the fields **Y**, **X** or **Z Unit**.

**b) Via the Replace or Convert Units dialog box**

1. Select the data sets that have a unit you want to replace. You can also select data sets that have different units.
2. Right-click with your mouse on the selected objects and choose **Change Units** from the context menu.
3. In the **Replace or Convert Units** dialog box enter the new units for the units that you want to replace.
4. For the **Action** select **replace**.
5. End the process by clicking **OK**.

**Converting Formulas, Analysis Objects or Data Link Objects into Data Sets**

1. Select all of the objects in the Object List that you want to convert into data sets.
2. Right-click with your mouse to open the context menu.
3. Select **Convert Into Data Set**.

---

**Note:** When you convert a formula into a data set, you lose the program code. To avoid this, you should lock the formula.

---

**Converting Analysis Objects or Data Link Objects into Formulas**

1. Select all of the objects in the Object List that you want to convert into formulas.
2. Open the context menu with the right mouse button.
3. Choose the **Convert Into Formula** command.

---

**Note:** When converting an analysis object into a formula, you lose the ability to set parameters via the Properties dialog box. You can, however, freely edit the resulting formula.

---

### 3.5 Data Sets

The data set is the data object in which FlexPro stores data that cannot be calculated as a formula. An example would be data that was imported from a file or entered manually. When importing multi-channel measurement data, FlexPro creates several data sets, one per measurement channel. Even if you enter data manually, you should create a separate data set for each measured quantity. It is especially important that all values in a data set have the same physical unit.

#### Data Editing Options

FlexPro offers you several options for directly editing data in data sets.




- If the data set to be edited is a data series or a data matrix, you can edit this in an open folder's data view. The data view allows editing across several data sets. This means that you can edit several data sets at the same time in a single data grid.
- In the data set window, you can edit each data set independently of its data structure. However, the data set window always only displays the data of an individual data set.
- If you are measuring the curves in diagrams using the cursors, you can edit the data under the cursor and between cursors directly.



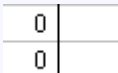
#### Data Set Window and Data View

The data set window and the data view function much like a spreadsheet in Excel or another spreadsheet program. You can edit them like in a spreadsheet, add, copy and move data and much more.



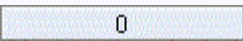

Data view allows you to work with the data of more than one data set as opposed to when using the data set window. All data sets with the data series and data matrix data structures in the open folder are displayed in one data grid.

The following table clarifies the user interface elements of the data set window:

Element	Purpose
	The upper left corner functions like a button; if you click on this button, the complete data set is selected.
	There are buttons on the left side that number the rows of the data set, starting with zero. By clicking on this button, you can select the complete row in the data set. The buttons on the left side are described as row headers.
	At the top you will find buttons identifying the component of the data set. If the component represents a data matrix, the components are numbered. By clicking on this button, you can select the complete column in the data set. The buttons at the top are described as column headers.

Element	Purpose
	These cells, which, depending on the structure of the data set, can be found at the bottom and/or on the right, act as placeholders for new values that can be inserted there.
	These cells identify areas where no input is possible. They result from the structure of the data set.
	Dividing lines provide visual separation of data set components, since all components of a data set are always displayed together in the data set window.

The following table describes the user interface elements of the data view:

Element	Purpose
	This button is located directly under the "Close" button. It selects all data.
	On the left-hand side are the buttons that number the rows starting with zero. By clicking on this type of button, you select the complete row in the data view. The buttons on the left side are described as row headers.
<b>Time (X)</b>	At the top you can see the name row, which shows the names and component identifiers of the data sets displayed in the data view. Selections in the name row correspond to the selections in the Object List. By dragging the mouse across the name row, you can rearrange the data sets in the data view.
	Underneath the name row you can see buttons identifying the columns of the data sets. If a data set represents a data matrix, the columns are numbered starting with zero. By clicking on this type of button, you can select the complete column. The buttons at the top are described as column headers.
	Empty cells at the bottom and/or on the right serve as fields for new values that can be inserted there. If necessary, the data set concerned will be enlarged, or a new data set is created.

In the data view and the Object List, you can merge several data series and/or data matrices into a single data matrix. To create a new data matrix, for instance, enter the data first as separate columns (data series) and then merge these into a matrix.

FPScript formulas can be used when entering values in the data view or in the data set window. To do this, the text that you input must start with '=', e. g., '=2 + 3' places the value 5 into the cell. You can also initialize one- and two-dimensional arrays; for instance, '=0#3#4' initializes a matrix with 3x4 elements with zeros.

You can also enter the values with a unit, such as '3 mA'. If the particular data set component does not have a unit, then the specified unit is assigned to all data.

Otherwise, the specified value is transformed to the unit of the data set. If this is not possible, an error message appears and the input value is discarded.

### Assigning Data Sets

The data view offers a particularly convenient way of assigning data sets. Simply specify for the individual data sets whether these are to represent the X, Y or Z component of an aggregate data structure. You can then determine from the order of the data sets which X or Z data sets are to be entered into the Y data sets headers as assigned data sets.

This also applies if you have several of these XY or XYZ arrangements, e.g. a X data set, to which three Y data sets are assigned and another X data set to which four Y data sets are assigned. The data view always arranges the Y data sets belonging to a certain X data set to the right of that X data set. You can always assign a data set identified as Y to another X data set by simply using your mouse to move this to position to the right of the other X data set.

Automatic assignment follows this pattern:

1. All Y data sets that follow an X data set without interruption are assigned this X data set.
2. A Y data set with a Z data set immediately to its right is assigned this Z data set.

Some examples of common data structures and their arrangement in the data view:

- **Time (X) Signal1(Y) Signal2(Y) Frequency (X) Spectrum1(Y) Spectrum2 (Y)**

**Signal1** and **Signal2** are time signals with a **Time** that is assigned as an X data set. **Spectrum1** and **Spectrum2** are frequency signals with a **Frequency** that is assigned as an X data set.

- **Sampling time(X) Measurements(Y) Trigger time(Z)**

The **Sampling time** and **Trigger time** data series are entered into the **Measurements** data matrix as an X or Z component. Together they form a signal series with a Z component.

- **XLocus (X) Measurements (Y) ZLocus (Z)**

The **XLocus** and **ZLocus** data series are locus coordinates for the data found in the **Measured values** data series. Together they form a space curve.

### Adding Components

You can also add an X or Z component with linearly increasing values to one or more data sets at a later time. This makes sense, for instance, when you know the sampling rate of the measured data, but the time information was not stored in the data file. You can access the relevant commands from the object list and data view.



## Working with Data Sets

### Creating a Data Set

#### In the Object List:

1. In the Folders window, select the folder where the new data set is to be placed.
2. Use your right mouse button to click on the Object List and open the context menu.
3. Select the **New** menu and then **Data Set**.
4. The Data Set Wizard appears to guide you through the remaining steps.

#### In the data view of the open folder:

- Enter data in an empty column. A data set with the data series data structure is automatically added that contains the data.

### Opening a Data Set

#### To open a data set in its own window:

- Double-click on its name in the Object List. If the data set is already displayed in the data view of the folder where it is located, FlexPro does not open a new window, but instead moves the data view to the foreground. Otherwise, FlexPro opens the data set in a new data set window.

#### To open a data set in the data view:

- Double-click on the file in **Folders window** containing the data set, and then click on the **Data** tab of the folders window. The data set will appear only if it has a data series or data matrix structure.

### Arranging Data Sets in the Data View

1. Select the data set you would like to move to a different position by clicking on its name.
2. Drag the data set to the required position and drop it there.

---

**Note:** If you change the order of the data sets in the data view, this will automatically change the assignments between the data sets. You will find out more under Assigning Data Sets.

---

### Selecting Data Sets in the Data View as Components

1. In the name row of the data view you select all those data sets which you would like to mark as a component.
2. Use your right mouse button to click on the selection and open the context menu.
3. In the context menu, point to **Component** and choose **X**, **Y**, **Z** or **None**.

### Changing the Data Type

1. In the data view, select a data set by clicking on its name in the header. You can skip this step in the data set editor.
2. Now choose **Change Data Type** from the **Data Set** menu or from the context menu.
3. On the **Data Types** tab in the data set's Properties dialog box, you can now select a different data type for the individual components of the data set.

### Merging Data Sets to a Data Matrix

1. In the name row of the data view or in the Object List, select all data sets you would like to merge into one data matrix.
2. Select **Merge into data matrix** from the **Data Set** menu or from the context menu.

---

**Note:** The resulting data set receives its name and further header information from the data set displayed on the far left. If the data sets have differing numbers of rows they might be filled with void values or empty strings.

---

### Adding an X or Z component

1. In the name row of the data view or in the Object List, select the data sets to which you want to add a component.
2. Select **Add X Component** or **Add Z Component** from the **Data Set** menu or from the context menu.

3. In the dialog box that now appears, choose whether the additional data component is to be added **as separate data series** or if it is to be included in the selected data sets.
4. Now add comments and the unit for the component and choose its data type.
5. Use the fields **Starting value** and **Increment** to set the linearly increasing or decreasing data for the new data series.

---

**Note:** If you use the command in data view and the option **As separate data series** is not enabled, then the data sets to which you have added a component will no longer appear in data view. Data sets with an aggregate data structure cannot be displayed in data view.

---

## Changing the Column Width

### Using the keyboard:

1. In the data view, select one or more data sets by clicking on the name in the header while holding down the CTRL key. You can skip this step in the data set window.
2. Choose the command **Change Column Width** from the **Data Set** menu or from the context menu.
3. Enter the desired column width.

### Using the mouse:

1. In the data view, select one or more data sets by clicking on the name in the header while holding down the CTRL key. You can skip this step in the data set window.
2. Move your mouse to the dividing line between two column headers.
3. Drag the dividing line to the desired position.

---

**Note:** This setting can be made for every data set individually and is saved together with the data set. Setting the width for individual components of a data set is not possible.

---

## Working with Data

### Navigating and Selecting Data

#### Using the mouse to navigate and select data:

Selection	Action
Single cell	Click on the cell or press an arrow key to navigate to the cell
Cell Range	Click on the first cell in the range and drag your mouse to the last row or hold down the SHIFT key and then use the arrow keys or Page Up or Page Down to move within the data grid. Alternatively, you can click on a cell and then use your mouse to navigate to the section of the data grid that contains the cell, which represents the opposite corner of the desired selection. Hold the SHIFT key down and click on this cell.
Entire Row	Click on the row header.
Entire Column	Click on the column header.
Adjacent Rows or Columns	Drag the mouse across the row or column headers.
All Cells	Click on the button in the top left corner of the data set window or press CTRL+A.

#### Using the keyboard to navigate and select data:

Key	Action
CURSOR KEYS	Move the cell selection in the corresponding direction, if possible. If you hold down the SHIFT key, the cells that you cross over are selected.
TAB, SHIFT + TAB	Moves the cell selection to the right or left, if possible.
ENTER KEY	Moves the cell selection down, if possible.
CTRL + HOME	Moves the cell selection to the top left corner of the data grid.
CTRL + END	Shifts the view onto the data grid so that the last row of the data grid is visible. The last cell of the last row is selected.
CTRL + LEFT	Moves the cell selection into the first column of the current row.
CTRL + RIGHT	Moves the cell selection into the last column of the current row.
CTRL + UP	Moves the selection to the first row of the current column.
CTRL + DOWN	Moves the selected cells to the last row of the current column.
SHIFT + LEFT	Extends the current selection by one column to the left.
SHIFT + RIGHT	Extends the current selection by one column to the right.

Key	Action
SHIFT + UP	Extends the current selection by one row up.
SHIFT + DOWN	Extends the current selection by one row down.
SHIFT + CTRL + LEFT	Extends the current selection to the first column.
SHIFT + CTRL + RIGHT	Extends the current selection to the last column.
SHIFT + CTRL + UP	Extends the current selection to the first row.
SHIFT + CTRL + DOWN	Extends the current selection to the last filled row.
SHIFT + CTRL + HOME	Extends the current selection to the upper left corner.
SHIFT + CTRL + END	Extends the current selection to the bottom right corner.
F2	Starts editing a cell value. The cell value is shown as being selected.
Any key	Replaces the text of the currently selected cell with the character entered. If the selected area was previously larger than one cell, it is reduced to the edited cell.

### Navigating to a particular cell:

1. In data view, click on any cell in the data set that you want to navigate to. You can skip this step in the data set editor.
2. Choose the **Go To** command from the **Data Set** menu.
3. In the dialog box that now appears, enter the coordinates of the cell to go to via the X index and, if applicable, Z index or via the X value and, if applicable, Z value.

### Changing and Adding Values

1. Before you can edit a data set, you have to open it. To do this, double-click on the data set in the Object List. If the data set is already displayed in an open folder's data view, the data view is activated and the data set is displayed. Otherwise, a data set window appears, allowing you to edit the data set.
2. Using the mouse or keyboard, move the cell selection to the cell you want to edit. To insert data, select an empty cell.

Key	Function
CURSOR KEYS	Move the cell selection in the corresponding direction, if possible. If you hold down the SHIFT key, the cells that you cross over are selected.
TAB, SHIFT TAB	Moves the cell selection to the right or left, if possible.

Key	Function
ENTER KEY	Moves the cell selection down, if possible.
CTRL + HOME	Moves the cell selection to the top left corner of the data grid.
CTRL + END	Moves the cell selection to the lower right corner of the data grid.
CTRL + LEFT	Moves the cell selection into the first column of the current row.
CTRL + RIGHT	Moves the cell selection into the last column of the current row.

3. Press the F2 key or start entering the desired value immediately.

4. Finish entering the value by pressing the ENTER key.

Your input is converted into the valid data type for the component and inserted into the data set.

Depending on the data type of the data set to which the selected cell in data view belongs, the following rules apply when you input values:

### Real Floating Point Values

For a void value, enter '?'.

### Complex Values

Both the real and imaginary parts of the complex value are entered in a cell. The data view displays complex numbers in the format '(real , imag)' or real+imagi and also accepts both input formats. When entering data in the format '(real , imag)', you can omit the parentheses. The two values must be separated by the list separator (usually ',') specified in the **Options** dialog box.

### Boolean Values

If you would like to enter Boolean values, you can either enter the texts 'TRUE' or 'FALSE', or specify 0 for FALSE and 1 for TRUE. The data view itself always displays Boolean values as words.

### Time Span Values

Enter these in the format 'D:H:M:S.ssss' or 'H:M:S.ssss'. Enter the fractional digits for seconds up to the desired resolution.

### Calendar Time Values

Here, FlexPro accepts the formats 'M/D/Y H:M:S.ssss', 'Y-M-D H:M:S.ssss' and 'D.M.Y H:M:S.ssss', . When entering data, you must always use these formats, even if the data is displayed in a different format.

**Note:** You can use FScript formulas to enter data. To do this, the text that you input must start with '=', e. g., '=2 + 3' places the value 5 into the cell. You can also initialize one- and two-dimensional arrays; for instance, '=0#3#4' initializes a matrix with 3x4 elements with zeros.

---

### **Adding Columns and Rows**

**To add one or more rows to the data set, which is displayed in the data set window, or to all data sets displayed in the data view:**

1. Select as many rows as you would like to insert by clicking and dragging the row headers.
2. Next, select **Insert Rows** from the **Data Set** menu or toolbar.

The selected number of new rows is inserted before the row selected.

**To add one or more rows to individual data sets in the data view:**

1. Select as many rows as you would like to insert as a cell range.
2. Next, select **Insert Rows** from the **Data Set** menu or toolbar.

The selected number of new rows will be inserted before the rows highlighted in the relevant data sets.

**To add one or more columns to the data set, which is displayed in the data set window, or in the data view:**

1. Select as many columns as you would like to insert by clicking and dragging the column headers.
2. Next, select **Insert columns** from the **Data Set** menu or toolbar.

The selected number of new columns will be inserted in front of the highlighted columns.

While in data view, if you had marked the columns in a data matrix, the number of columns will increase. Otherwise, new data sets will be created which start out empty.

---

**Note:**

- No rows or columns can be added for scalar values.
- No columns can be added for scalar values, data series, signals and space curves.
- With signal series, columns can only be inserted in the range of the Y component.

- For signal series with a two dimensional X component, at least two columns (for X and Y) always have to be inserted.
- 

### Deleting Columns and Rows

**To delete one or more rows from the data set, which is displayed in the data set window, or from all data sets displayed in the data view:**

1. Select the rows to be deleted by clicking and dragging on the row headers.
2. Next, select **Delete Rows** from the **Data Set** menu or toolbar.

The rows selected are then deleted from the data set(s).

**To delete one or more columns from the data set that is displayed in the data set window or in data view:**

1. Select the columns that you want to delete by clicking and dragging on the column headers.
2. Next, select **Delete Columns** from the **Data Set** menu or toolbar.

The columns selected are then deleted from the data set.

---

#### Note:

- You cannot delete the only column or row if there are only scalar values.
  - No columns can be deleted for signals and space curves.
  - The X component and, if present, any Z component cannot be deleted for signal series.
  - At least two columns (X and Y) must always be deleted in the case of signal series with a two-dimensional X component.
- 

### Copying Data via Drag-and-Drop from the Object List

You can use the mouse to add entire data sets to a data set in the data set window or in data view:

1. Select the data set to be added from the Object List.
2. Drag the selected data set to the point in the data set window or data view where you want to insert the data. A gray frame appears in locations where you are allowed to insert data. The frame size depends on the size of the source data set. By moving your mouse, the frame moves in the data grid. If you move the mouse to the edge of



the data set window, the visible section is moved as far as possible in the direction of the movement.

3. Move the gray frame to the desired location and then release the left mouse button. The data from the data set is then copied to the selected location. Any previously existing data will be overwritten.

### **Copying Data via the Clipboard**

1. Select the desired data set in the Object List or the desired area in a data set window or in data view and run the **Copy** command.
2. In the data set window or in data view, move the cell marker using the mouse or keyboard into a gray cell. Depending on the data structure of the edited data set, these type of cells are located under and/or next to the values that are already present. For scalar values, no new data can be entered.
3. Use the **Paste** command. The values from the clipboard are inserted from the insertion point heading right and/or downward. Any previously existing data will be overwritten. If the structure of the data in the clipboard does not match the structure of the data set in the data set window or the insertion point, the **Paste** command will not be available.

### **Copying Data via Drag-and-Drop from a Data Set Window or the Data View**

You can insert ranges from data sets into the data set in the data set window using the mouse:

1. In the data set window of the source data set or in data view, use your mouse or keyboard to select the range that you would like to transfer into the target data set.
2. Move the mouse to the edge of the selection. The shape of the mouse cursor changes from the cell cursor to an arrow cursor.
3. Hold down the left mouse button and drag the range to the position in the data view where you would like to copy the data. A gray frame appears at the positions where insertion is possible. The frame size corresponds to the size of the selected range. By moving your mouse, the frame moves in the data grid. If you move the mouse to the edge of the data set window or data view, the visible section is moved as far as possible in the direction of the movement.
4. Next, release the left mouse button at the desired location. The selected data is then copied to the selected location. Any previously existing data will be overwritten.

### Filling Columns with Values

1. In data view, select the empty columns you want to fill with values by dragging the mouse over the column headers or selecting a cell range. If you select entire columns, the filling range will be set to the longest data set in the data view.
2. Choose the command **Fill Column with Values** from the **Data Set** menu or from the context menu. You can request further help in the dialog box that appears.

### Calculating Values in Columns

1. Select the columns you would like to calculate by dragging the column headers or selecting a cell range. If you select whole columns, the range to be calculated is adjusted according to the longest data set in the data view.
2. Select **Calculate Values in Columns** from the **Data Set** menu or from the context menu. In the dialog box that appears, you can request further help.

## 3.6 Excel Workbook

FlexPro can open your Microsoft Excel workbooks directly and display them in a window. To display an Excel workbook, Excel is launched in the background. As soon as you click on the Excel workbook, the complete Excel user interface with all menus and toolbars is at your disposal and you can work with them as you normally would. Excel 97 or later must be installed on your machine so that you can work with Excel workbooks.

You can embed Excel workbooks into the FlexPro project database or create links to external Excel files. You can save an Excel workbook embedded in the project database into an external file at any time.

FlexPro allows you to process the data in your Excel workbooks directly. To do this you have to create Excel data links. These can be handled in the same way as regular data sets. For the data and the header information, such as comments, physical units, etc., you can create links with individual cells or cell ranges in the Excel workbook. The Excel data link therefore combines several ranges from the Excel workbook into one complete data set.

FlexPro handles Excel workbooks like folders in the project database. You store the Excel data links to an Excel workbook in this folder. You can, of course, also store any number of other objects in this type of Excel workbook folder. You can activate an Excel workbook like a normal FlexPro folder.

## Excel Data Link

FlexPro uses Excel data link objects to access the data of an Excel workbook in the project database. You can assign a cell or a cell range from the Excel workbook in the Excel data link for every element that makes up a FlexPro data set. The Excel data link reads all the information necessary for one FlexPro data set from the Excel workbook. This applies not only to data, but also to all header information such as comments, physical units and names of physical quantities.

You can use this type of Excel data link just like a normal data set. Whenever FlexPro requests the data or other information from the Excel data link, it refers back to the assigned Excel workbook. Elements for which there is no information in the Excel workbook can also be entered directly in the link.

An Excel data link can have all of the data structures supported by FlexPro. You can, for instance, specify the cell ranges for the X and Y component of a signal. Usually, you create several data links to an Excel workbook. Often one column in the Excel workbook corresponds to one Excel data link.

Excel data links always have to be placed in the Excel workbook folder, whose associated Excel workbook you want to access. Furthermore, you cannot use Excel data links to exchange data between FlexPro and a separately launched Excel application. This requires that you create an OLE data link.

## Working with Excel Workbooks

### Creating an Excel Workbook

1. In the Folders window, select the folder where you want to place the object.
2. Select **Excel Workbook** from the **New** menu or from the **FlexPro Objects** toolbar. The Excel workbook will now be created and opened.

---

**Note:** Since Excel workbooks are treated like folders in FlexPro, you will find the same **List** and **Data** tabs in an Excel workbook window as in the window of an open folder.

---

### Opening an Excel Workbook

- Double-click on the name of the workbook in the Folders window or right-click on its name in the Object List and select **Open** from the context menu.

### Importing an Excel Workbook

1. Right-click with your mouse in the Object List. To import the Excel workbook into a different folder, use your right mouse button to click in the Folders window on the folder that is to receive the data.
2. Choose the **Import** command.
3. In the dialog box that now appears, select from the **Excel workbook** file format from the **File Type** toolbar. Select the file(s) you would like to import.
4. Select the option **Create links** if you want to create a link to the Excel file(s). Otherwise, a copy of the Excel workbook(s) is placed in the project database.

### Exporting an Excel Workbook

1. In the Folders window or the Object List, use your right mouse button to click on the Excel workbook.
2. Choose the command **Export**.
3. In the **Export** dialog box that appears, select the folder into which you want to export the file and give the file a name.

### Switching between External Storage and Embedding

1. In the Folders window or the Object List, use your right mouse button to click on the Excel workbook.
2. Choose the command **Properties**.
3. In the **Properties** dialog box, switch to the **Excel Workbook** tab.
4. Select the preferred storage format.
5. If you want to store the table externally, you need to specify a complete path name for the file.

### Troubleshooting Excel Communications

To allow you to edit Excel workbooks and to access Excel data, FlexPro must work closely with Microsoft Excel. This interaction will run smoothly if you observe the following:

1. Make sure that Excel is completely and correctly installed on your computer. FlexPro requires Microsoft Excel version 97 or later. If Excel is not installed, Excel workbooks cannot be created or opened. If the installation is incomplete or faulty, this can lead to random effects such as error messages on opening Excel workbooks or the incomplete

display of data.

2. If you are working with Excel workbooks, you should not start Excel as a separate application at the same time. This could result in a situation where Excel cannot respond to queries from FlexPro because Excel may be blocked by an open dialog box. This could result in error messages when accessing data and the incomplete display of Excel workbooks in FlexPro.
3. If you should encounter an error in Excel communications, shut down FlexPro and make sure that Excel is not running in the background. You can call up the Windows Task Manager by pressing CTRL, ALT and DELETE at the same time. If an unnecessary EXCEL.EXE process is displayed, you should shut it down. Restart FlexPro.

## Working with Excel Data Links

### Manually Creating an Excel Data Link

If you are creating an Excel data link manually, you can specify the cell or cell range in the Excel workbook for each data element. Do the following:

1. If necessary, open the Excel workbook folder and then double-click on the name of the associated Excel workbook to which you want to link.
2. Select **Excel Data Link** from the **New** menu or **FlexPro Objects** toolbar. The Excel data link will now be created. A dialog box appears, allowing you to assign cells or cell ranges for the individual elements.
3. In the **Create Excel Data Link Object** dialog box, place the cursor on the field with the cell or cell range you want to assign.
4. In the Excel workbook, select the cell range or the cell you want to assign to the field selected. FlexPro then enters the cell reference of the selection into the field, e.g. **Sheet1!B3:B11**.
5. Once you have set the cell ranges for the data, you should test whether your selection returns a valid result. This is displayed at the bottom of the dialog box.

---

**Note:** If comments and physical units are not included in your Excel workbook, you can also enter these manually or leave the fields blank. If the units in brackets are attached to the table titles, e. g. **Time[s]**, you can still select the cell. FlexPro automatically extracts only the unit and ignores the name and the brackets.

---

### Creating Excel Data Links via the Clipboard

1. Select a cell range in the Excel workbook containing data to which you would like to create one or more links. If above the data there is a title row with column names and, if applicable, attached physical units, you should include this row into the selection.
2. Use the **Copy** command to place the selected data on the clipboard.
3. Now switch to the Excel workbook folder's list view by clicking on the **List** tab at the bottom of the window. If the Excel workbook folder is selected in the Folders window, you can also click in the Object List to activate this.
4. Use the **Paste** command. A dialog box appears where you can specify how the Excel data links are to be created.
5. Check all settings and request further help in the dialog box, if necessary.

### Creating Excel Data Links Using Drag-and-Drop

1. Make sure that the Excel workbook folder is not only open, but also selected in the Folders window so that you can see its contents in the Object List.
2. Select a cell range in the Excel workbook containing data to which you would like to create one or more links. If above the data there is a title row with column names and, if applicable, attached physical units, you should include this row into the selection.
3. Move the mouse to the edge of the selection. The shape of the mouse cursor changes from the cell cursor to an arrow cursor.
4. Hold down the left mouse button and drag the range down to the Object List. A plus symbol now appears on the mouse arrow. If this symbol does not appear, then you should check if the correct folder is selected in the folder window.
5. Release the mouse button. A dialog box appears where you can specify how the Excel data links are to be created.
6. Check all settings and request further help in the dialog box, if necessary.

### Analyzing and Presenting Excel Data

You can also analyze or present cell ranges directly by creating a FlexPro analysis object, a diagram or a table. FlexPro then creates the necessary Excel data links in an intermediate step:

1. Select a cell range in the Excel workbook with data that you would like to analyze or visualize. If above the data there is a title row with column names and, if applicable,

- attached physical units, you should include this row into the selection.
2. Select the type of the newly created object from the **New** menu on the **FlexPro Objects** toolbar.
  3. A dialog box appears where you can specify how the Excel data links are to be created.
  4. Check all settings and request further help in the dialog box, if necessary.
  5. Confirm with **OK**. An object of the type selected is created in the folder displayed and, if necessary, a wizard will appear first to assist you in setting the object properties.

### 3.7 Unit Manager

The value of a (measurement) quantity is expressed as a product of a number and unit. FlexPro features a Unit Manager that handles direct import, analysis and presentation of these types of quantities. The customizable Unit Manager in FlexPro is based on the SI unit system and includes the following features:

- Compatibility check and automatic adjustment when calculating quantities of different units.
- Percentage calculation with % and ppm units.
- Ability to select output unit separately for display of quantities in diagrams and tables.
- Support for additional popular unit systems (Gaussian unit system, US unit system).
- Open system architecture in relation to unknown units.
- Manual and automated unit correction during import..
- Ability to define and use custom units.
- Support for the ISO 8000 standard International System of Quantities (ISQ).

#### The SI Unit System

The International System of Units, "Système International d'Unités" (SI) was introduced in 1960 and is the most widely used unit system for physical quantities. It is managed and internationally standardized by the International Bureau of Weights and Measures, "Bureau International des Poids et Mesures" (BIPM).

The unit system defines seven **base units** through physical measuring methods:

Base unit	Symbol	Base quantity	Symbol	Dimension symbol
Meter	m	Length	l, x, r, etc.	L
Kilogram	kg	Mass	m	M

Base unit	Symbol	Base quantity	Symbol	Dimension symbol
Second	s	Time	t	T
Ampere	A	Electric current	I, i	I
Kelvin	K	Temperature	T	
Mole	mol	Amount of substance	n	N
Candela	cd	Luminous intensity	I <sub>v</sub>	J

Additional **coherent SI units** are **derived** exclusively from these base units. The coherent SI unit of electrical resistance, the ohm with the unit symbol  $\Omega$ , is, for instance, uniquely defined by the relation  $\Omega = \text{m}^2 \text{kg s}^{-3} \text{A}^{-2}$ , which follows directly from the definition of the physical quantity. **Coherent** in this context means that no factors other than one occur in the product of powers.

For some of these derived **units, special names and symbols**, such as Pascal (Pa), are in turn used; others are assembled from the existing names and symbols, such as the Newton meter (N m). The **notation of the symbols** is case sensitive. Symbols derived from a proper name start with a capital letter, such as Newton (N) or Pascal (P). Symbols where this is not the case are written in lower-case, e.g. second (s).

Some measurement quantities span several orders of magnitude. To avoid extra large or small numeric values when displayed, you can place an **SI prefix** in front of the unit symbol. A unit defined this way, however, is no longer coherent, since the prefix introduces a factor that is not equal to one. Each SI prefix corresponds to a particular power of 10, so for instance the millimeter (mm) is equal to  $10^{-3}$  m, or the kilovolt (kV) is equal to  $10^{+3}$  V.

The appearance of a unit as a product of powers of the base units defines the **SI dimension** of the respective quantity. This SI dimension is important when checking units for compatibility. Compatible, i.e. interconvertible units, such as km/h and m/s, always have the same SI dimension, as in the example  $\text{M}^1 \text{T}^{-1}$ . The reverse is unfortunately not the case, i.e. units of different kind can have the same SI dimension. A familiar example of this are the units Newton meter (N m) for rotational speed and joule (J) for energy. Both have the dimension  $\text{L}^1 \text{M}^1 \text{T}^{-2}$ .

Some derived quantities, in particular quantities defined as ratios of quantities of the same kind, have an SI dimension in which all exponents are equal to zero. The product of powers is thus equal to one, and therefore these quantities are denoted as from the **dimension one** or as **dimensionless**.

The exponents of the base units usually are whole numbers in the range from  $\pm 4$ . In certain cases, however, fractions may also appear as exponents. Therefore, this results in rational exponents, for instance, when some Gaussian unit system units are displayed in SI base units.



## Calculating and Converting Quantities

When making calculations with quantities, you need to account for more than just their numeric values. Thus, for instance,  $1\text{ N} + 400\text{ mN}$  neither results in  $401\text{ N}$  nor in  $401\text{ mN}$ , but instead results in  $1.4\text{ N}$ . Even the statement  $1\text{ N} > 400\text{ mN}$  would provide an incorrect result if purely accounting for the numeric values. Other operations are physically useless and should therefore be recognized as such, e.g.  $1\text{ V} + 1\text{ A}$ . On the other hand, for other calculations a new unit appears for the result, e.g.  $6\text{ m} / 2\text{ s} = 3\text{ m/s}$  or  $5\text{ m}^2 = 25\text{ m}^2$ . The unit in which the data is output should also be independent of the unit in which it was acquired, e.g. a speed may be measured in  $\text{m/s}$ , but output in  $\text{km/h}$ . The FlexPro Unit Manager covers all of these scenarios. All FPScript mathematical operators as well as all built-in analysis functions not only account for the values of quantities, but also their unit, consisting of the SI dimension, factor, offset and unit symbol. FlexPro is able to compare units prior to calculation, determine the correct unit for the result of a calculation, recognize nonsensical operations, i.e. incompatible units, and transform units for output.

## Unit Types

FlexPro makes a distinction between units with known and unknown SI dimensions. Units with an unknown SI dimension cannot be transformed, which means in particular that when using addition and subtraction as well as comparable operations, the units of both operands must match exactly. Multiplication, division and exponentiation, however, can be used without this limitation, and the symbol of the resulting unit is also formed correctly. The FPScript Debugger displays units with an unknown SI dimension in quotation marks, e.g. "people/h".

## Expandable Unit Collections

FlexPro supports a basic set of units from which you can derive new units through multiplication, division and exponentiation. In addition, you can also use unit collections for the US unit system, the Gaussian unit system and a set of other popular non-SI units, which you can enable as another option.

You can expand on this set of units by registering your own custom units and specifying their SI dimensions. FlexPro manages a unit table in the current project database, in the personal template database, in the global template database and in a number of shared template databases, the locations of which you can specify on the **Units** tab of the FlexPro **Organizer** dialog box. You and your colleagues can store units that you want to share in shared template databases. The unit tables contained within are read by FlexPro when the program is launched.

You can also add units with unknown SI dimensions to the unit tables. This will ensure

that FlexPro does not convert the unit or calculate it with other units.

When translating a unit, FlexPro first searches through the unit tables in the order mentioned above and attempts to find an exact match. If no entry is found, the unit is being decomposed into its element. For elements that are not recognized as SI unit symbols, FlexPro searches through the tables again.

Example: FlexPro first searches through the unit tables for the entire unit "Vrms A". If this cannot be found, it will be decomposed into the elements "A" and "Vrms". "A" is the known SI unit Ampere and is accepted as such. "Vrms" is not known to FlexPro, which is why FlexPro searches through the tables again to find it. If an entry is found, the symbol is accepted. The SI dimension specified in the entry is then multiplied with the SI dimension of "A". Otherwise, the entire unit "Vrms A" is viewed as an unknown unit.

### Unit Monitoring Modes

You can configure one of the following unit monitoring modes for each project database:

- **None**  
FPScript ignores all units and performs calculations with numeric values only. Some analysis objects determine the unit of the result using the units entered in the header data of the arguments. In this mode, FPScript is compatible with the previous versions of FlexPro up to version 8.
- **Tolerant** (preset for new project databases)  
FPScript performs calculations with quantities, i.e. values with a unit, adjusts units to each other during processing, and checks units for compatibility before processing. When output to diagrams, curves of an axis are set to the same unit before being displayed. This mode also permits units whose SI dimension is unknown to FlexPro. These types of units must match exactly when processing two data sets. In addition, in this mode, compound units without spaces or multiplication signs between the element units are permitted, such as "Nm" instead of correctly as "N m" or "N·m".
- **Moderate**  
Specifies the Tolerant setting with the limitation that compound units without spaces or multiplication signs between the element units are not permitted.
- **Strict**  
Specifies the Moderate setting with the additional limitation that units whose SI dimension is unknown to FlexPro are not permitted.

---

**Note:** The default setting for newly created project databases is set on the **Unit Manager** tab of the FlexPro **Options** dialog box. You can configure the setting for the open project database on the tab of the same name in the project database **Properties** dialog box.

---

## Physical Quantities

The ISO 8000 series of standards define a large number of physical quantities and their SI units from different domains. FlexPro provides you with assistance when entering a unit and physical quantity for a data set. When you specify a unit, only the physical quantities compatible with this unit are displayed in the physical quantity list box. Conversely, when you choose a physical quantity, only the units compatible with this physical quantity are displayed in the unit list box.

---

**Note:** You can specify which domains FlexPro should include on the **Unit Manager** tab of the FlexPro **Options** dialog box.

---

## Unit Correction during Data Import

Many measurement systems either do not store the physical unit in the data files at all or they store it in an incorrect notation. A typical mistake, for instance, is to use the symbol S instead of the correct symbol s for the Second unit. In the SI unit system, "S" stands for the unit "Siemens", which is used for conductance. Other examples include RPM instead of 1/min, grdC instead of °C, sec instead of s, and so on. Often, they do not even distinguish between the unit name and symbol, as in the case of VOLT or Volts instead of V, for instance.

This is why FlexPro can provide you with a list of all units found in the data as you start importing it so that you can review and correct them as necessary. During this process, you can add your corrections to a correction table so that FlexPro can make these corrections automatically during subsequent imports.

FlexPro manages a unit table in the current project database, in the personal template database, in the global template database and in a number of shared template databases, the locations of which you can specify on the **Units** tab of the FlexPro **Organizer** dialog box. You and your colleagues can store units that you want to share in shared template databases. The unit tables contained within are read by FlexPro when the program is launched. For automatic unit correction, FlexPro uses the correction tables in the order mentioned above.

## Internal Representation of Units

To calculate quantities accurately and convert them into other units, FlexPro stores the following additional quantity attributes in addition to the numeric value(s).

- The unit symbol as a string,
- The denominator and numerator of the exponents of the seven SI base units,

- A factor, which comprises all prefixes used in the symbol or represents the factor between the existing unit and SI base unit (e.g. 1 min = 60 s), and
- An offset, which represents a distance between the unit within the quantity and the respective SI base unit (e.g. 1 °C = (1 + 273,15) K).

FlexPro can use this data to convert a quantity from the existing unit into a compatible SI unit:

$$SIValue [SIUnit] = Value [Unit] \cdot Factor \left[ \frac{SIUnit}{Unit} \right] + Offset [SIUnit]$$

with

$$SIUnit := m^{\frac{N_1}{D_1}} kg^{\frac{N_2}{D_2}} s^{\frac{N_3}{D_3}} A^{\frac{N_4}{D_4}} K^{\frac{N_5}{D_5}} mol^{\frac{N_6}{D_6}} cd^{\frac{N_7}{D_7}}$$

The inverse operation can be used to convert a quantity in the SI unit to another unit with a known SI dimension. Both steps together result in the following formula for converting a quantity from one unit to another:

$$Value_2[Unit_2] = Value_1[Unit_1] \cdot \frac{Factor_1}{Factor_2} \left[ \frac{Unit_2}{Unit_1} \right] + \frac{Offset_1 - Offset_2}{Factor_2} [Unit_2]$$

In addition, FlexPro can check units for compatibility prior to calculation. Units are considered compatible if their SI dimensions match. This check is not 100% foolproof (the Newton meter and joule units, for instance, have the same SI dimension), but it catches the majority of errors.

You can obtain the two vectors with the denominators and numerators of the seven exponents as well as the factor and offset of a unit in FlexPro using the SIUnits function. The expression SIUnits(1 km/h) provides, for instance, a list with the following four elements:

Scale = 0.277777777777778

Offset = 0

ExpNum = {1, 0, -1, 0, 0, 0, 0}

ExpDenom = {1, 0, 1, 0, 0, 0, 0}

The 1 km/h expressed in SI base units is 1000 m / 3600 s = 0.277777777777778 m<sup>1/1</sup> s<sup>-1/1</sup>.

For non-existing base units, the exponential numerator is equal to zero. The upshot is that for a **dimensionless unit**, all exponential numerators are always equal to zero. In this case, the exponential denominator is interpreted by FlexPro in a special format in order to distinguish between dimensionless units of varying types. The unit of the solid angle, the steradian (sr), for instance, has the following exponents:

ExpNum = {0, 0, 0, 0, 0, 0, 0}

ExpDenom = {2, 0, 0, 0, 0, 0, 0 }

These are interpreted as  $\text{m}^2/\text{m}^2$ . Similarly, the unit of the angle, the radian (rad), provides the following exponents, which are interpreted as  $\text{m}/\text{m}$ :

ExpNum = {0, 0, 0, 0, 0, 0, 0 }

ExpDenom = {1, 0, 0, 0, 0, 0, 0 }

Since FlexPro compares the denominator and numerator for compatibility when comparing two units, it recognizes that the operation  $1 \text{ rad} + 1 \text{ sr}$  is not permitted, even though both have the same SI dimension.

## References

- Bureau International des Poids et Mesures (2008). The International System of Units, 8th Edition. [http://www.bipm.org/en/si/si\\_brochure/](http://www.bipm.org/en/si/si_brochure/)
- ISO 80000 standards 1 through 13.

## Working with the Unit Manager

### Setting Up the Unit Manager

#### a) Setting up the Unit Manager for the open project database

1. In the FlexPro **File** menu, select **Properties**.
2. In the **ProjectDatabase Properties** dialog box click on the **Unit Manager** tab.
3. Here you can change the Unit monitoring settings, percentage calculation and unit collections to be activated as desired. Call up additional help in the dialog box if necessary.

---

**Note:** Your changes to the settings for FPScript formulas in the project database will take effect immediately. You will have to check and, if necessary, make changes to the FPScript code.

---

#### b) Setting up the Unit Manager default settings for new project databases that you create.

1. In the FlexPro **Tools** menu, select **Options**.
2. In the **Options** dialog box click on the **Unit Manager** tab.
3. Here you can change the Unit monitoring settings, percentage calculation and unit collections to be activated as desired. Call up additional help in the dialog box if

necessary.

**c) Selecting the categories of measurement quantities to be displayed in list boxes.**

1. In the FlexPro **Tools** menu, select **Options**.
2. In the **Options** dialog box click on the **Unit Manager** tab.
3. Select the categories relevant for your work in the **Offer quantities from the following categories** list.

**Checking Imported Units**

Depending on the settings of the Units option for data import, the **Verify Units** dialog box appears. Proceed as follows to check and, if necessary, to correct the imported units.

1. First specify in the **Action** list box whether you want the units to be edited automatically or manually. Select **Edit units** to check the units individually and, if necessary to correct them.
2. Point the mouse to an imported unit in the first column of the list to obtain a summary of the data sets that use this unit.
3. Now check the displayed SI dimension. If **Unknown** appears here, then you should correct the unit. Even if an SI dimension is displayed, you should check it carefully to ensure that the unit is interpreted as expected. If the unit is correct, you can skip the next item.
4. Click on an incorrect unit in the column **Replace With** and specify the correct unit symbol. Confirm your entry by pressing the ENTER key. Recheck the SI dimension.
5. If nothing appears in the **Physical Quantity** column, then you should specify a quantity. To do this, click on the empty field in this column and choose one of the quantities displayed. You can also enter a name of your own choice.
6. If the corrections made are to be carried out automatically for future imports, select a storage location in the **Save changes** list box. Your replacements will then be added to the selected table when the dialog box is closed. You can use the **Edit selected table** button to check the current content of the table.
7. Click **Close** to close the dialog box. The selected action is then carried out.

### Registering Custom Units

1. In the FlexPro **File** menu, select **Properties**.
2. In the **ProjectDatabase Properties** dialog box that appears, click on the **Unit Manager** tab.
3. Click on the button **Edit table of custom units** on the tab to open the **Table of Custom Units** dialog box.
4. From the **Table** list box, choose the table that you want to edit.
5. Click on **Insert item** or press the INSERT key.
6. In the **Unit** column of the list, enter the symbol of the new unit using the correct case-sensitive. You cannot use symbols that are already assigned to other units.
7. Now click on the field to the right of the symbol and enter the representation of the unit in SI units. Please note the following rules for entering a unit:
  - You can use any SI units you want during input and not just the seven base units, such as "N" or "W".
  - Enter "1" for a dimensionless unit.
  - Unit symbols are case-sensitive, which means, for instance, that "S" is the symbol for the unit "Siemens" and "s" is the unit for seconds.
  - Use a space for multiplication, e.g. "N m" instead of "Nm".
  - Use the '^' character for exponents, e.g. "m^2".
  - Use the '/' character for division, e.g. "m/s^2".
  - Use parentheses correctly, e.g. "kg/(kW h)".
  - You can specify an SI prefix or a factor, e.g. "mm" or "0.001 m".
  - You can specify an offset, e.g. "K-273.15".
8. If you want to allow the use of an SI prefix for your new unit, click in the **Allow SI Prefix** column and select **Yes**.
9. Click **Close** to accept your changes.

---

**Note:** You cannot edit the unit tables in the shared template databases directly. For instance, to add a unit to a shared template database, you first have to register it in the current project database or the personal template database as described here and then copy it to the shared template database, as described in Organizing Units.

---

### Editing Tables for Unit Correction during Data Import

1. In the FlexPro **File** menu, select **Properties**.
2. In the **Project Database Properties** dialog box that appears, click on the **Unit Manager** tab.
3. Click on the button **Edit table for correcting units during data import** on the tab to open the **Data Import Unit Correction Table** dialog box.
4. From the **Table** list box, choose the table that you want to edit.
5. Click on **Insert item** or press the INSERT key.
6. In the **Unit** column of the list, enter the symbol of the new unit using the case-sensitive.
7. Now click on the field to the right of the symbol and enter the symbol that will replace this unit. If the unit you enter is known, a check mark will appear at the beginning of the row. Otherwise, an exclamation mark will appear.
  - Unit symbols are case-sensitive, which means, for instance, that "S" is the symbol for the unit "Siemens" and "s" is the unit for seconds.
  - Use a space for multiplication, e.g. "N m" instead of "Nm".
  - Use the '^' character for exponents, e.g. "m^2".
  - Use the '/' character for division, e.g. "m/s^2".
  - Use parentheses correctly, e.g. "kg/(kW h)".
  - You can specify an SI prefix or a factor, e.g. "mm" or "0.001 m".
  - You can specify an offset, e.g. "K-273.15".
8. Click **Close** to accept your changes.

---

**Note:** You cannot edit the unit tables in the shared template databases directly. For instance, to add a unit to a shared template database, you first have to register it in the current project database or the personal template database as described here and then copy it to the shared template database, as described in Organizing Units.

---



## Organizing Units

Use the **Organizer** dialog box to delete units from the custom units table or from the unit correction table during data import or to copy units from one database to another.

### Opening the Organizer dialog box

1. In the FlexPro **Tools** menu, select **Organizer**.
2. In the **Organizer** dialog box click on the **Units** tab.

### Opening the current project database or a template database for organization

- Choose a template database in one of the **Items available in** list boxes.

### Opening a shared template database or any other database for organization

1. Click on one of the **Close Project Database** buttons to close the database currently displayed in the list.
2. Click on **Open Project Database** and select any project database on your hard disk or on the network.

### Copying units from one project database to another

1. First, in the **Table** field, choose whether you want to edit the custom units table or the unit correction table for data import.
2. Now select the units that you want to copy and click **Copy**.

### Deleting units from a project database

1. First, in the **Table** field, choose whether you want to edit the custom units table or the unit correction table for data import.
2. Now select the units that you want to delete and click **Delete**.

### Checking Imported Units

Depending on the settings of the Units option for data import, the **Verify Units** dialog box appears. Proceed as follows to check and, if necessary, to correct the imported units.

1. First specify in the **Action** list box whether you want the units to be edited automatically or manually. Select **Edit units** to check the units individually and, if necessary to correct them.

2. Point the mouse to an imported unit in the first column of the list to obtain a summary of the data sets that use this unit.
3. Now check the displayed SI dimension. If **Unknown** appears here, then you should correct the unit. Even if an SI dimension is displayed, you should check it carefully to ensure that the unit is interpreted as expected. If the unit is correct, you can skip the next item.
4. Click on an incorrect unit in the column **Replace With** and specify the correct unit symbol. Confirm your entry by pressing the ENTER key. Recheck the SI dimension.
5. If nothing appears in the **Physical Quantity** column, then you should specify a quantity. To do this, click on the empty field in this column and choose one of the quantities displayed. You can also enter a name of your own choice.
6. If the corrections made are to be carried out automatically for future imports, select a storage location in the **Save changes** list box. Your replacements will then be added to the selected table when the dialog box is closed. You can use the **Edit selected table** button to check the current content of the table.
7. Click **Close** to close the dialog box. The selected action is then carried out.

## 3.8 Importing Files

FlexPro offers you various data import options, which are presented below:

### Importing Data from Binary Files

FlexPro supports binary file formats from many different measuring systems and data acquisition programs and can import these directly. A generic import filters for binary data is also available. For most data formats, you can choose whether you want to import individual channels from a file or whether to import the entire file. FlexPro can read several files of different formats in a single step. Optionally, a folder of the same name can be created in the project database for every file to be imported.

### Importing Data from Text Files

You can import text files using the Text Import Wizard. The wizard offers two import methods. All settings that you specify while importing a file can be saved as a new file type. This allows you to call up any number of files with the same structure at a later stage without having to use the wizard again. You can obtain additional information by accessing help in the Text Import Wizard.

## Data Link Object

Not only can you copy the data from binary and text files into the FlexPro project database, but you can also create links to the original data. Links are created via data link objects, which read the data of a channel from a file, if necessary. You can use the data link object as you would a data set or a formula. Data link objects are special formulas that have parameters you can set using a dialog box.

If the file you want to import contains several data sets, such as a multi-column text file, then an individual data link object is set up for each data set. This object only reads the section assigned to it from the file. You cannot to set up the data link objects manually. FlexPro automatically creates these when you select the **Create links** option during import. Read...File FPScript import functions are available for various data formats that read out a channel from the file. These are used in the formulas of data link objects.

Since data link objects maintain the link to the file to be imported, you can therefore exchange data dynamically between the application that writes to the file and FlexPro. You can transform data link objects into data sets at any time. When updating, the formulas of data link objects are only considered invalid if the external file has been changed since last read in. This ensures that unnecessary re-calculations are avoided.

Some data link objects can read comments and units from the files to which they are linked. You can activate this feature by leaving the corresponding input fields empty in the Properties dialog box of the import object.

## Importing Data from ODBC Data Sources

FlexPro can import data from ODBC data sources.

Open Database Connectivity (ODBC) is an international standard for manipulating relational data using the SQL syntax. ODBC offers a standard interface for accessing relational data sources using what are known as ODBC drivers, which exist for various project database systems.

ODBC is part of MDAC (Microsoft Data Access Components), which is supplied in different versions with the different Windows operating systems. You can view a list of the ODBC drivers available to you by invoking the **Control Panel**, going to **Administrative Tools** and clicking on the **Drivers** tab under **ODBC Data Sources**. You can download the latest version of MDAC from Microsoft's website.

## Importing Data Using Data Explorer

If the **Data Explorer** option is installed, FlexPro can index all your data on the hard drive. You can use the **Data Explorer** to search through the data and import it into the Object List using drag-and-drop, for instance.

### **Importing Data Using DataQuery**

If the **Data Explorer option** is installed, you can use the **DataQuery** object to directly access all data that FlexPro has found and indexed on your computer. The object directly imports the data to which the query applies and passes it as a list.

### **Importing Data from ASAM ODS Data Sources**

ASAM ODS is a standard used in the automotive industry for data management. If the ASAM ODS Data Import option is installed, FlexPro can create links to and import data from ASAM ODS conforming data sources.

### **Importing Data from from ISO/TS 13499 MME Data Sets**

The ISO/TS 13499 MME format is widely used in the automotive industry as a multimedia format for crash tests. FlexPro can import data, images and videos from ISO/TS 13499 MME data sets. Images and videos are added as media objects that remain linked to the original files.

### **Exchanging Data with Excel**

The integration of Microsoft Excel into FlexPro makes it very easy for you to exchange data between FlexPro and Excel. You can find out more under Excel Workbook and Excel Data Link.

### **Importing Data via OLE**

OLE (Object Linking and Embedding) is a standard protocol for data exchange in Windows. To import data using OLE, the application that is to supply the data must have an OLE server interface.

In FlexPro, you use OLE data links to access data from the OLE server, e.g. Microsoft Word. You can use these links like normal data sets within FlexPro. Whenever the data in the OLE data link is accessed, the request is passed on to the server application.

The server application must offer the data in text format so that FlexPro can convert the data. If you transfer a single column, the OLE data link provides this column in the form of a data series. If you link several columns, then they are provided as a data matrix. Values specified with "#NV" are imported as void values.

## Importing Data

### Importing Data from Files of Other Programs

#### a) Using the Import command:

1. Right-click with your mouse in the Object List.
2. From the context menu, choose **Import**.
3. In the dialog box that now appears, select from the **File Type** list the file format that you want to import. Select **All Files** if you want to import several different file types. Select **Text Files** or **Binary Files** if the specific format is not listed.
4. Select the file(s) you would like to import.
5. Now choose the procedure to be used for handling physical units when importing data.
  - **Copy without changing** The known imported units are copied without being changed. If the Unit Manager is set to **Strict**, unknown units will be deleted. Otherwise, these units will also remain unchanged when copied.
  - **Correct automatically** All imported units are corrected automatically using the unit correction tables in the project database and template databases. If the Unit Manager is set to **Strict**, remaining unknown units will be deleted.
  - **Check if unknown** All imported units are first corrected automatically using the unit correction tables in the project database and template databases. If there are any remaining unknown units, a dialog box will appear where you can check the units and correct them, if necessary.
  - **Check all** All imported units are first corrected automatically using the unit correction tables in the project database and template databases. A dialog box will then appear where you can check and, if necessary, correct all units. You can also edit the descriptions of the physical quantities for the data in the dialog box.
6. If you want to import the complete file(s), select the option **Fast import of complete files**. If you want to import individual channels from a file, deselect the option **Fast import of complete files**.
7. If you want to link the data instead of copying it, select the option **Create links**.
8. Some import filters can import the time information from signals as physical time or as calendar time (date and time). Use the option **Use calendar time** to choose between the two.
9. Select the option **Import as signals** if the time information of signals is to be integrated into the data sets. Otherwise, the data sets and the time information are stored as separate data series.

10. Select the option **Mark data sets as read only** if you want to protect imported data from being changed inadvertently.
11. Click **Open** to open the **Import** dialog box for the selected format.
12. Depending on the import format, follow the instructions of the wizard or request additional help in the dialog box.
13. Depending on how the **Units** option is set, the **Verify Units** dialog box appears. For information on how to use this dialog box, see Checking Imported Units.

---

**Notes:**

- The availability of the options described above depends on the import format selected.
- If you import read-only data sets, then the data editor and any other operations that could change the data are disabled. For instance, you will not be able to eliminate or interpolate any outliers with the cursors.
- If you carry out a fast import for text data, the text data must be arranged in columns. Each column should have a title to which the physical unit can, as an option, be appended in square brackets. The decimal symbol must correspond to the Regional and Language Options set in the Windows Control Panel.

---

**b) Importing data using drag-and-drop:**

You can import data very quickly if you use drag-and-drop to drag the data directly from Windows Explorer to the Object List.

1. Select the files in Windows Explorer that you want to import.
2. Drag the data from Windows Explorer to the Object List. If you want to link the data instead of copying it, you hold down the CTRL and SHIFT keys while you drop the files into the Object List. A linking symbol should appear on the cursor.
3. All imported units are first corrected automatically using the unit correction tables in the project database and template databases. If units remain that are unknown to FlexPro, the **Verify Units** dialog box appears. For information on how to use this dialog box, see Checking Imported Units.

---

**Note:** Not every import filter supports linking. Importing using drag-and-drop corresponds to the **Fast import of complete files** and **Create new folder for each file** options of the **Import** command.

---

## Importing Data via the Clipboard

### a) Importing data into the FlexPro project database

You can import values or ranges of values from other applications directly into the project database. The data will then be stored as new data sets in the **Object List**. The pre-condition for this is that the external application must be able to provide the values in text format in the clipboard or by using a drag-and-drop operation. To import values from Microsoft Word, for instance, do the following:

1. Start Microsoft Word and open the document where the data you want to import into FlexPro is located.
2. Select the text or the table to be transferred and run the **Copy** command from the **Edit** menu.
3. Switch to FlexPro and click in the **Object List**.
4. Use the command **Paste** from the **Edit** menu.
5. A dialog box now appears where you can specify how the data is to be imported. Select the options you wish to use and close the dialog box by clicking on **OK**.

...or use drag-and-drop.

1. Start Microsoft Word and open the document where the data you want to import into FlexPro is located.
2. Select the text or table to be transferred.
3. Move the mouse cursor onto the selected area. Hold down the left mouse button and the CTRL key and move the mouse onto the **Object List** while keeping the CTRL key and mouse button pressed. When the mouse pointer turns to a + symbol, this indicates that the data is being copied.
4. Now release the left mouse button and then the CTRL key.
5. A dialog box now appears where you can specify how the data is to be imported. Select the options you wish to use and close the dialog box by clicking on **OK**.

---

### Note:

- You can select single values or one or more columns of values and import them in a single operation. If you just selected two columns, you can import them as a signal, for instance. In this case, the first column specifies the X values and the second specifies the Y values of the signal. If you have selected multiple columns, you can import them as a data matrix or as separate data series. A data matrix is a two-dimensional data set that is suitable for a matrix of values, for instance. If you import several data series, then a separate data set is created for each column selected. Here

you can specify if a specific column is to be used as an X data set. The name of this data series is then entered as an assigned data set in the headers of all Y data series.

- If you import data series or signals, the first line of the selected text may contain the captions of the columns. FlexPro then uses them as the names of the data sets to be created. As an option, you can append the physical unit set off by square brackets, such as "Temperature [°C]".
- 

### b) Importing data into an open data set or the data view of an open folder

You can import values or ranges of values from other applications into an open data set or the data view of an open folder. Also in this case, the values must be provided in text format. To import values from Microsoft Word, for instance, into a data set window, do the following:

1. Start Microsoft Word and open the document containing the data you want to import into FlexPro.
2. Select the text or the table to be transferred and run the **Copy** command from the **Edit** menu.
3. Switch to FlexPro into the data view or the data set window where you want to insert the values.
4. Select the cell that is to represent the top left corner of the range or field to be inserted.
5. Use the command **Paste** from the **Edit** menu.

...or use drag-and-drop.

1. Start Microsoft Word and open the document containing the data you want to import into FlexPro.
2. Select the text or table to be transferred.
3. Move the mouse cursor onto the selected area. Press the left mouse button and move the mouse onto the data view or the data set window where you want to insert the values. The area of insertion is indicated by a rectangular frame.
4. Release the left mouse button after you have selected the insertion point. The values are moved from Word into the selected area or are copied there if you keep the CTRL key pressed as you release the left mouse button.

---

**Note:** When importing data from external applications, the import process fails if FlexPro cannot convert the data into the data type of the import target area. In addition, empty lines between the values are imported as void values. Empty lines at the beginning are ignored.

---



### Importing Data from ODBC Data Sources

Importing data using ODBC is similar to the process of importing data from files:

1. Right-click with your mouse in the Object List.
2. From the context menu, choose the **Import** command.
3. In the dialog box that now appears, click on the **ODBC** button.
4. Select a data source from which you want to import data in the **Select Data Source** dialog box, or create a new data source.
5. The **ODBC Data Import** dialog box appears where you can select the columns to import from the tables present in the data source. If necessary, call up Help in the dialog box by clicking on '?'.
6. The import process starts when you press **Copy** or **Link**.

---

**Note:** Some data formats, such as Microsoft Excel workbooks (extension .XLS), Microsoft Access databases (extension .MDB) and also ODBC data source files (extension .DSN), can also be selected and imported directly as files.

---

### Importing Data Using OLE

In this section, you will create an OLE data link. The process depends on the server application from which you would like to import the data. Here we describe the process using the popular word processor Microsoft Word.

1. In Word, open the document from which you would like to import the data.
2. Select a text or a table with numerical data.
3. Use the command **Copy** from the **Edit** menu to copy the OLE object to the clipboard.
4. In the Folders window, select the folder where the OLE data link is to be inserted.
5. Use the command **Paste Special** from the **Edit** menu to insert the OLE data link.
6. In the **Paste Special** dialog box, select **Link**.
7. Click on **OK**.

## Importing Data Using Data Explorer

### "Indexed Data" View

---

**Note:** The option **Show All Indexed Data** or a search filter needs to be selected.

---

Only a file or channel can be selected in the tree view, whereas multiple channels can be selected at the same time in the list view.

In contrast to File Explorer, this view allows importing of complete files as well as individual channels.

#### a) Importing Data via the Clipboard

1. Select the file or channels in Data Explorer that you want to import.
2. Right-click with the mouse and from the context menu choose **Copy**.
3. Hover the mouse cursor over the object list and from the context menu choose **Paste**.

#### b) Importing data using drag-and-drop

You can also import data if you use drag-and-drop to drag the data directly from Data Explorer to the Object List.

1. Select the file or channels in Data Explorer that you want to import.
2. Drag the data from Data Explorer to the Object List. If you want to link the data instead of copying it, you hold down the CTRL and SHIFT keys while you drop the files into the Object List. A linking symbol should appear on the cursor.

#### c) Importing data using Data Explorer commands

1. Select the file or channels in Data Explorer that you want to import.
2. Right-click with the mouse and from the context menu choose **Copy to Project Database**. If you want to link the file instead of copying it, choose the following from the context menu: **Create &Link(s) in Project Database**.
3. The list view also includes the following commands: **Copy All Channels of Selected File(s) to Project Database** or **Create All Channels of Selected File(s) As Link(s) in Project Database**. This allows you to select one or more channels and import all channels of the associated files.

### "File Explorer" View

---

**Note:** To do this, select **Show File Explorer**.

---

Importing data from File Explorer is handled similarly as importing from Windows Explorer.

Only a file can be selected in the tree view, whereas multiple files can be selected at the same time in the list view.

**a) Importing Data via the Clipboard**

1. Select the files in File Explorer that you want to import.
2. Right-click with the mouse and from the context menu choose **Copy**.
3. Hover the mouse cursor over the object list and from the context menu choose **Paste**.

**b) Importing data using drag-and-drop**

You can also import data if you use drag-and-drop to drag the data directly from File Explorer to the Object List.

1. Select the files in File Explorer that you want to import.
2. Drag the data from File Explorer to the Object List. If you want to link the data instead of copying it, you hold down the CTRL and SHIFT keys while you drop the files into the Object List. A linking symbol should appear on the cursor.
3. All imported units are first corrected automatically using the unit correction tables in the project database and template databases. If units remain that are unknown to FlexPro, the **Verify Units** dialog box appears. For information on how to use this dialog box, see Checking Imported Units.

---

**Note:** Not every import filter supports linking. Importing using drag-and-drop corresponds to the **Fast import of complete files** and **Create new folder for each file**.

---

### Importing Data Using DataQuery

You can import several data sets as a list by querying the external data (indexed data).

---

**Note:** To do this, indexing must be enabled. Querying external data is only possible when the **Data Explorer Option** is installed. Select the search criteria for the data query so that the result does not contain too many channels/data objects, since this can have a strong impact on the processing speed.

---

#### Importing sample data

Import the ultrasonic signals from the file samples provided. The folder C : \Users\Public must have already been indexed in order to proceed.

1. Add a new DataQuery data object.
2. On the **Options** tab, select **Search in Indexed Files**.
3. Select **Channels / data objects in folder**: C:\Users\Public
4. Select **Include subfolders in search**.
5. Define a search criterion **Name is ultrasonic signal**.
6. Select **Values of found channels / data objects as list** as the result.

#### Single channels / output data objects

1. Use the command **Run data query** to test the data query. Use the **Select Items** button to list the results of the data query in a tree and select or deselect individual channels/data objects from there.
2. Select only one of the found ultrasonic signals.
3. Make sure the option **Output selected elements from previous data query** is selected.

## 3.9 Data Export

### Exporting Data in a File

FlexPro offers you the option of exporting data sets, formulas and related objects in diverse formats.

When saving in CSV or text format, you have the option of exporting several objects below one other or side-by-side to a single file. You can also select different file formats and specify whether the result or formula text is to be output for the formulas. CSV stands for **Comma Separated Values**. The values are separated in this format with the

list separator specified in the Windows **Control Panel** under **Regional and Language Options**.

Only signals and signal series with two signals are suitable for export as an audio file. Signals are treated as mono signals, signal series with two signals are treated as stereo signals, and signal series with more than two signals are treated as multi-channel audio signals. The data's X component must correspond to one of the following sampling rates supported by the audio file format: 11025 Hz, 22050 Hz or 44100 Hz.

### FlexPro Data Exchange Format

The FlexPro data exchange format is similar to that variant of the text format, where the data of several data sets are stored side-by-side and with all of the header information. Here, however, an additional row with a data format identifier is then output. Regardless of the system settings, floating point values are always output with a point as the decimal symbol. Calendar times are also always output in the same format, regardless of the system settings. This format has the following characteristics:

- The structure of the data format is simple, in that it is relatively easy to integrate corresponding export and/or import routines into programs that want to exchange data with FlexPro on a file basis.
- The complete header information including parameter lists is stored. This means that no information is lost when data is exchanged.
- The data format corresponds to a single table in ANSI code. The tab character (09 Hex) serves as column separator. The table is completely square, meaning that all rows have the same number of columns. This is why the format is particularly suitable for exchanging data with spreadsheet programs.
- The data format is independent of the system's Region and Language Settings. Floating point values are always output with a decimal point, and calendar settings are always output in the format MM/DD/YYYY HH:MM:SS in local time.
- Formulas can either be stored like data sets with data or purely as formulas without data. During import, data sets are created for the first case, and in the second case, formulas are created.
- Data with linearly increasing values can be stored using the two parameters called starting value and increment, which means that in this case the corresponding data columns may remain empty.

The following table describes the data format in detail:

Row	Contents
1	Version identifier: FlexPro Text Data File Version 4.0 at UTC+/-TimeShift The TimeShift number following UTC specifies the delay in hours relative to UTC (Coordinated Universal Time). This allows calendar times in the file to be converted back

Row	Contents
	to UTC.
2	Names of the data sets and formulas contained inside. The names are shown in the first column of the applicable data set. If a data set has several data columns, several column separators can come between its name and that of the neighbor to the right.
3	String with data structure identifier for each data set. The format of the identifier is <data structure> <number of columns> <number of rows> <Y data type> <Y formatter> <X data type> <X formatter> <Z data type> <Z formatter>. The following keywords are used for the data structure: Scalar, DataSeries, DataMatrix, Signal, SignalSeries, SignalSeriesZ, SignalSeriesX, SignalSeriesXZ, SpaceCurve and Formula. The X following SignalSeries specifies that it has a two-dimensional X component, i.e. a separate X column exists for every Y column. The Z following SignalSeries specifies that it has a Z component. The number of rows and columns is only shown if the number of dimensions of the data structure requires this. The Formula keyword specifies that only a formula but no data has been stored for the corresponding name. In this case, data types and dimension lengths are not shown. Only the three formatters follow the keyword. This type of formula without data is stored like a space curve with three columns, which, however, remain empty. The following keywords are used for the data types of the individual components: Integer16, Integer32, Integer64, FloatingPoint32, FloatingPoint64, ComplexInteger16, ComplexInteger32, ComplexInteger64, ComplexFloatingPoint32, ComplexFloatingPoint64, Boolean Value, String, DateTime, DurationHMS, DurationDHMS. If a component contains a data series with linearly increasing real values, the keyword is followed by the starting value and increment for the row in curly brackets, e.g. FloatingPoint64{0 0.1}.. The starting value and increment always have to be specified as numerical values, meaning that calendar time values and durations are not allowed. In this case, the corresponding data columns may remain empty, since they are ignored during import. The formatters correspond to the entries on the <b>Formatting</b> tab of the FlexPro <b>Properties</b> dialog box of the data sets. These are output as strings and must be set off by quotation marks. Quotation marks appear as doubled, since the complete data structure identifier is also output as one large string. Data types and formatters for X and Z components are only output if they exist in the corresponding data structure.
4	Comments for the individual components of the data sets and formulas. Output occurs in the order X, Y, Z, with the X and Z component only shown if they are present. The Z component is only output here for space curves and formulas. For signal series with Z components, this is not a column, but a row and its comment is output separately.
5	The hyperlinks assigned to the data sets and formulas.
6	Details on the individual components of data sets and formulas. For the X and Z component, the specifications for Row 4 apply.
7	The description of the physical quantity for the individual components of the data sets and formulas. For the X and Z component, the specifications for Row 4 apply.

Row	Contents
8	The lower limit ranges for the individual components of data sets and formulas. For the X and Z component, the specifications for Row 4 apply. The values are formatted as specified under 27.
9	The upper limit ranges for the individual components of data sets and formulas. For the X and Z component, the specifications for Row 4 apply. The values are formatted as specified under 27.
10	Absolute times for the individual components of the data sets and formulas. For the X and Z component, the specifications for Row 4 apply. The values are formatted as specified under 27.
11	Creation date and time of the data sets and formulas are in the format MM/DD/YYYY HH:MM:SS. Output is in local time.
12	Date and time of the last modification to data sets and formulas.
13	Author of the data sets and formulas.
14	Origin of the data sets and formulas.
15	Component selection: None, Y, X, Z Specifies whether a data set contains all data or is to be interpreted as the component of an aggregate data structure.
16	Name of the assigned X data set in case a data set is marked as a Y component.
17	Name of the assigned Z data set in case a data set is marked as a Y component.
18	A list of parameters for each data set and each formula. For each parameter, an expression in the format <data type> <name>{<value>}[<unit>] is output. The data type is coded as specified under 3. The values are formatted as specified under 27. For complex values, however, real and imaginary components are separated with a space, and strings are always set off by quotation marks. These are then doubled, since the whole list is seen as one large string. The unit, if present, is placed in brackets following the value when output.
19	The comments of the Z components of all signal series with a Z component. See also Row 4.
20	The units of the Z components of all signal series with a Z component. See also Row 4.
21	The names of the physical quantities of the Z components of all signal series with a Z component. See also Row 4.
22	The lower range limits of the Z components of all signal series with a Z component. See also Row 4.
23	The upper range limits of the Z components of all signal series with a Z component. See also Row 4.
24	The absolute times of the Z components of all signal series with a Z component and formulas. See also Row 4.

Row	Contents
25	The FScript codes of formulas.
26	The data of the Z components of all signal series with a Z component. Formatting as specified for Row 27.
27	The data of the data sets and formulas. For complex data, the real part is output first and the imaginary part is output to a second column. For signals and signal series, output occurs in the order X, Y. For space curves, the order is X, Y, Z. Signal series with a two-dimensional X component are output in the order X0, Y0, X1, Y1 ... Xn-1, Yn-1. Void values are output as #NV. Floating point values are output with a decimal point. For Boolean values, TRUE or FALSE is output. Calendar time values are shown in local time as MM/DD/YYYY HH:MM:SS. If applicable, seconds are shown with fractional digits.

All rows are always output, even if they contain nothing but column separators.

The number of columns of the data table not only depends on the number of data sets to be exported, but also on their data structure and data type.

When strings are stored, they are placed in quotation marks if they contain at least one quotation mark or any control character. If a string is placed in quotation marks, all quotation marks contained within are doubled. Control codes within the string are not changed. This means that carriage returns included in a string must not be interpreted as a new row of the FlexPro data table.

The current version of the format is 4.0. The following table shows the version change history:

Version	Changes
1.0	Initial version.
2.0	The data types Integer64 and ComplexInteger64 were added.
3.0	The rows 7, 8, 9, 21, 22 and 23 specified above were added in which the data range limits and names of physical quantity of the individual components are stored. For parameters with units, they are now appended in brackets.
4.0	The rows 10 and 24 specified above were added in which the absolute times of the individual components are stored.

## Exporting Data via OLE

OLE (Object Linking and Embedding) is a standard protocol for data exchange in Windows. To export data via OLE, the application that is to receive the data must have an OLE client interface. FlexPro can provide the data in CSV and text formats.



## Exporting Data via the Clipboard or via Drag-and-Drop

You can export individual data values or ranges from a data set into other applications using the clipboard or via drag-and-drop. FlexPro provides the data in CSV and text formats.

## Exporting Data

### Exporting Data Objects as a File

1. Select all of the objects in the Object List that you want to export to the file. Select a folder in order to export all data objects contained within it (not supported by all formats).
2. Right-click with your mouse to open the context menu.
3. Select **Export**.
4. Select the target folder and specify the file name for the file.
5. Under **File Type**, select the format for the file.
6. Select any export options for the file.

### Exporting Data Objects via OLE

The way the link is inserted depends on the external application into which the data is to be inserted. The process described here is therefore an example of exporting to a Microsoft Excel spreadsheet.

1. Select a data set, a formula or a related object in the Object List.
2. Use the command **Copy** from the **Edit** menu to copy the OLE object to the clipboard.
3. In Excel you open the table into which you want to insert a link to FlexPro files.
4. Select the cell that corresponds to the upper left corner of the column or field to be inserted.
5. Use the command **Paste Special** from the **Edit** menu.
6. In the **Paste Special** dialog box, select the format **CSV** and select the option **Link**.

### Exporting Data via the Clipboard or via Drag-and-Drop

Inserting the values depends on the external application into which the data is to be inserted. The process described here is therefore an example of exporting to a Microsoft Excel spreadsheet.

1. Open the data set from which you want to export the values.
2. In the data set window or in an open folder's data view, select the values that you want to export.
3. Use the command **Copy** from the **Edit** menu to copy the selection to the clipboard.
4. Use Excel to open the spreadsheet into which you want to insert the values.
5. Select the cell that corresponds to the upper left corner of the column or field to be inserted.
6. Use the command **Paste** from the **Edit** menu.

or

1. Open the data set from which you want to export the values.
2. Select the values that you want to export from the data set window.
3. Move the mouse cursor to the edge of the selected area. The cell cursor turns into an arrow cursor. Press the left mouse button and drag the mouse to the Excel worksheet into which you would like to insert the values. The insertion area is shown by a rectangle.
4. Release the left mouse button after you have selected the insertion point. The values are copied from FlexPro into the selected area.

## 3.10 Data Explorer Option

The **Data Explorer** option gives you the ability to display the contents of files without having to import the data itself. You can also preview single channels as well as statistical quantities, such as maximum, minimum and mean.

Use the search function for a simple way to search for data set properties, preselecting what you need for file import. The results can then easily be imported into FlexPro.

You can use the **DataQuery** object to find and browse through data on your hard drive and then further process this data directly without having to import it.

### File Indexing

File indexing in FlexPro sifts through your hard drive data and prepares an index database, which is used by **Data Explorer** and the **DataQuery** object to find data.

FlexPro not only stores all header information in the database, such as units, comments and parameters, but also information about the data itself, such as the value count, sampling rate, mean or a thumbnail for the data preview.

Before you can use Data Explorer, you need to set up File Indexing tab. To do this, specify the folders to be searched on your hard drive and the data formats to be included in the search. You can also specify when FlexPro should perform indexing and at what time intervals the index database should be updated.

If you enable file indexing, FlexPro will close, but then FlexPro will remain open in the background so that it can index the files. This can be identified by the FlexPro icon that appears in the Windows Taskbar system tray. If you right-click on the icon, a menu appears which you can use to quit FlexPro, for instance.

### **Data Explorer Views**

Data Explorer features two basic views:

The **Data Explorer** view is similar to Windows Explorer. All files and folders on your hard drive and on the network are displayed. You can import files using drag-and-drop or copy them to the Object List using the Data Explorer clipboard.

The **indexed files** view shows you only the files that are detected and indexed by FlexPro. In this view you can see individual data channels in the measurement data files. If you click on a data channel, detailed information will be displayed in the Properties window and in the preview.

You can view both views as a list or as a tree in a hierarchical structure.

### **Preview and Properties**

If you use your mouse to point to a channel displayed in Data Explorer, a popup window will with detailed information will appear. If you select the channel in Data Explorer, a preview and properties will be displayed.

### **Search Filters**

Search filters allow you to specify which data to display in Data Explorer. In addition to the default search filter, you can also define additional search filters which FlexPro can store in the project database or in a template database. You can combine multiple search criteria in a single search filter. For instance, you can search for all data sets whose names start with "Axial", have a unit of  $\text{m/s}^2$  and a maximum exceeding 10.

## Using Data Explorer

### Setting Up File Indexing

1. In the **Tools** menu, click **Options**.
2. In the **Options** dialog box, click on the **File Indexing** tab.
3. Select the desired options.

---

**Note:** You can also open the **Options** dialog box using the **Options** button on the **Data Explorer** window toolbar.

---

### Mode

Determines the indexing mode for data files on the hard drive.

- Select **No indexing** to disable indexing.
- Select **Index in idle mode** to enable indexing as long as FlexPro is not used interactively. Use this option if indexing interferes with your work in FlexPro.
- Select **Always index** to enable indexing to run in the background even while you are working with FlexPro. Use this option if you want to complete the indexing as quickly as possible.

Several indexing options are available:

- **Restart:** Restarts the indexing, i.e. the program searches through all the data again from the beginning, but does not delete the index database.
- **Delete and create new index:** Discards the index database and restarts indexing from the beginning.

In the **Indexing repeat rate** field, specify how often the program should search for new files to index on the hard drive.

Use the option **Run file indexing when Windows starts up** to specify whether indexing should start automatically when Windows starts up.

### File Types To Index

Here you choose which file types known to FlexPro should be included when indexing, or select the **Index all supported file types** option.

In the **Excluded file types** field, specify the file types (e.g. \*.exe; \*.sys) that should be ignored when indexing.

### Locations To Index

Here you specify the hard drive folders in which the program will search for data files. Use the **Subfolders** column to specify whether to search through specific subfolders as well.

### Changing the Data Explorer View

In **Data Explorer** you can view all indexed data or search results.

---

**Note:** Indexed data can only be displayed, if data indexing has been activated and the files have already been indexed.

---

### Displaying File Explorer

- Select the option **Show File Explorer** from the **Search** dropdown list box in the toolbar of the Data Explorer window.

### Displaying All Indexed Data

- Select the option **Show All Indexed Data** from the **Search** dropdown list box in the toolbar of the Data Explorer window.

### Displaying Search Results

- Select the option **Find...** from the **Search** dropdown list box in the toolbar of the Data Explorer window to select pre-defined search filters or to define a new filter. Pre-defined search filters appear in the **Search** dropdown list box and can be selected there.

### Changing the View

- From the **Change View** dropdown list box in the toolbar of the Data Explorer window, you can choose between **Tree view** and **List view** .

### Viewing Channel Properties

- You can display different properties for the indexed channels in the list view. Right-click with the mouse on the column title to view several properties such as name, unit and comments which you can display or hide.

### Updating the View

You can update the view in **Data Explorer** by clicking on the **Update** button in the toolbar of the Data Explorer window.

---

**Note:** An update may be required if the indexed data has changed.

---

### Searching in Data Explorer

In **Data Explorer** you can set up search filters for indexed data and perform a search using a number of search criteria.

---

**Note:** To do this, file indexing must be enabled.

---

### Setting Up Search Filters

1. Select the option **Find...** from the **Search** dropdown list box in the toolbar of the Data Explorer window, or click on the **Search** button in the toolbar of the Data Explorer window.
2. Specify where to save the search filter. For a one-time search that will not be saved, you can select **Data Explorer** and define a standard filter. Otherwise, you have the option of saving the search filter in a personal template database or the open project database.
3. Add a new search filter and give it a name.
4. Under "Search Criteria", specify the properties that you want to find.

### Selecting Search Filters

Pre-defined search filters appear in the **Search** dropdown list box and can be selected there.

## 3.11 Data Queries

DataQuery is a data object used to search for objects with shared properties and to return their data. Data on the hard drive (indexed files) or in the project database can be queried.

For instance, it is possible to search for all objects that have a particular name or a particular unit and return their data as a list.

You have the option of concatenating the results of multiple data queries. This is useful if you want to perform several data queries using shared search criteria. You can do this by having a data query return the keys of the found data objects and then using them in another data query.

To be able to use the data query option, indexing must be enabled.

- To query data on the hard drive, File Indexing must be set up.
- To query data in the project database, data indexing for the project database must be enabled.

---

**Note:** Querying data on the hard drive is only possible when the Data Explorer Option is installed.

---

## Working with Data Queries

### Setting Up a Data Query

1. In the Folders window, select the folder in which you want to add the new data query.
2. Right-click with your mouse in the Object List to open the context menu.
3. Point to **New** and in the menu that appears, click on **Data Query**.
4. A DataQuery object is added with a unique name to the displayed folder.
5. Click on **?** in the dialog box to obtain additional help with configuration options.

### Concatenating Data Queries

1. Add a new DataQuery data object.
2. On the **Options** tab, select **Search in Project Database** or **Search in Indexed Files**.
3. Select **All indexed channels / data objects**
4. Define one or more search criteria.
5. Select **Keys of found channels / data objects** as the **result**.
6. Add another DataQuery data object.
7. On the **Options** tab, select **Search in Project Database** or **Search in Indexed Files**.
8. Select **Channels / data objects with keys** and select the first data query with the

found keys.

9. Define one or more search criteria.

10. Select as the result **Values of found channels / data objects as list.**

11. Repeat steps 6 through 10 for additional data queries that build upon the first data query.



## 4 Analyzing Data with Cursors

FlexPro allows you to measure curves in 2D and 3D diagrams and videos using cursors.

More specifically, the following options are available:

- Show one or two cursors using different display formats.
- Free and bound cursors.
- Harmonic cursors for measuring secondary harmonics.
- Show values under the cursors, their differences and custom coordinates.
- Zoom in on or out of curves and scroll through them.
- Move to individual frames in videos.
- Animate a cursor to play back curves.
- Translate curves in 2D diagrams.
- Navigate to peak values.
- Select individual points or ranges of a curve.
- Edit the values under the cursors.
- Remove outlying values from curves.
- Form new data sets from sections of curves.
- Dimension curves.

You can use the cursors in 2D and 3D Diagrams, in Media objects, in Documents and in Worksheets. If you use the cursors in a worksheet or document, you can simultaneously measure multiple diagrams and analyze videos that you have recorded synchronously with physical quantities. However, the worksheet is generally better suited than the document as an environment for using the cursors, because it takes better advantage of the screen. While the cursors in the worksheet are always active, you first have to activate them in the diagram, media or document.

The toolbar used for working with cursors is usually located at the bottom right of FlexPro's main window. Most cursor commands can be found there. Additional options are available on the **Cursors** tab of the Properties dialog box related to the object controlling the cursors.

---

**Note:** If you use the cursors in a worksheet or a document, then these are controlled by the worksheet or document and not by the diagrams. You must therefore adjust the settings on the **Cursors** tab in the Properties dialog box of the worksheet or document and not in the Properties dialog box of the diagram.

---

### Cursor Types

You can use one or two cursors for each diagram. The leading cursor is displayed as a solid line and is used to measure curves. The origin cursor is shown as a dashed line. It specifies the origin for the calculating differences.

FlexPro offers you free and linked (bound) cursors. You can move a free cursor anywhere on the plane of the diagram, while a linked cursor always follows the points of the curve. Free cursors are particularly suitable for measuring parts of the curve where there are no points or for measuring distances to a free reference position. You individual curves of a diagram from cursoring. The cursor can then no longer be positioned on the curves, and their values are not displayed in the coordinates window.

You can display the cursors as a vertical line or as a cross-hair. You can also move the cross-hair cursor vertically with your mouse.

FlexPro offers you harmonic cursors that are specifically designed for measuring secondary harmonics in relation to a primary harmonic in spectral signals. If you activate this option, in addition to the active cursor, a maximum of ten more cursors are displayed, which are positioned at multiples of the current X cursor position. You can specify the spacing between harmonic cursors in octaves, where one octave corresponds to a factor of two.

Media objects support only a cursor that represents the time of the currently displayed frame. Use the horizontal scrollbar so you can move to an image and thus move the cursor.

### Cursor Synchronization

If you use the cursors in a worksheet or document with multiple diagrams and/or videos, FlexPro can synchronize the cursors between the individual objects. Synchronization can be carried out via the X index, i.e. via the number of the active value in the curve or frame in the video. Alternatively, FlexPro can synchronize the cursors via the X value. In a video, this is the time corresponding to the displayed frame; in a curve, this is the X value of the active point.

FlexPro can also synchronize the cursors while you play a video or audio signal. In this case, the cursors in the diagrams are animated and move in sync with the video or audio signal over the curves. Conversely, you can move a cursor in a diagram to display the relevant frame in the video, e.g. for a position at which you identified a point of interest in a measurement signal.

### Zooming and Scrolling

You can zoom into any section of the diagram's plane or show the area between the two

cursors as spread out in order to take a closer look at interesting signal segments. All processes that change the section displayed in the diagram achieve this by manipulating the axis scale end values in the diagram. In addition, automatic scaling on the axis concerned is disabled. This modification takes place directly on the axis in the diagram concerned. If you edit a linked diagram with cursors, zooming therefore affects all documents where you have inserted a link to the diagram. You can undo any step in the process of zooming and scrolling or restore the diagram to its original state. If you move the cursor on a curve you have zoomed into, it might leave the visible section. In this case, you can determine whether the image section should be automatically expanded or moved.

### **Animating a Cursor**

You can animate any cursor, which means you can let it run automatically over a curve. If the curve under the cursor is a time signal, the animation takes place in realtime or at a multiple rate of this speed. When viewing an image you have zoomed into, the section of the image may be automatically expanded or moved when the cursor leaves it. The signal then scrolls across the screen.

### **Markers**

When measuring curves, you can select individual points or ranges, for instance, to highlight prominent points in the curve. When setting a marker, you can determine whether the marker is to be displayed in the current view of the diagram or in all views that display the diagram. In 3D surface diagrams, you can select any number of rectangular areas of the surface.

You can mark individual points with a **text marker**, which you can place either at the top of the diagram or right at the highlighted point. In the text marker, you can show any text in which you can embed the coordinates of the selected points using fields.

You can select sections of curves and rectangular surface areas using a **range marker**. One of these types of range markers can also be used to assign text, but this is not displayed in the curve.

An **image marker** places an image at the highlighted position. If you have recorded videos synchronously with measurement data, FlexPro can automatically take a still image for the highlighted X position from an assigned media and use it for the image marker. Alternatively, you can specify an image file from which the image is to be imported.

You can copy the markers to the clipboard as a new data set and then save them in the project database.

### Changing Data Set Values

If the curve where the cursor is located obtains its data from a data set, then you can manipulate its values directly using the cursors. The cursors offer you the option of changing a single value or the range between the cursors to a different value, interpolating them or setting them to void. The first option is ideal for the visual adjustment of a calibration characteristic curve, for instance, and the two others are good for removing outliers from a data set. Setting outliers to void is generally preferred over interpolation, since in this case no false data is being introduced. However, the advantage of interpolation is that the resulting data cannot contain void values, and thus calculations can still be made using algorithms, such as the Fourier transform.

### Translating curves

You can translate the curves of 2D diagrams using the cursor, e.g. to compensate for time lag when measuring. In this case, the curves are only visually translated. The base data remains unchanged. The offset is entered as **Y offset** and **X offset** on the **Data** tab of the curve.

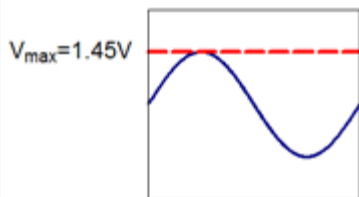
### Coordinates Window

In the coordinates window, the coordinates of the cursors and their differences are displayed. To do this, you can specify whether only the coordinates of the active diagram or the coordinates of all diagrams in a document or worksheet are to be displayed. You can copy the contents of the coordinates window to the clipboard as a static picture or into a diagram, document or worksheet as a dynamic object. In addition to the coordinates window, FlexPro displays an information window with the scaled coordinates of the current mouse pointer position if you move the pointer over the diagram.

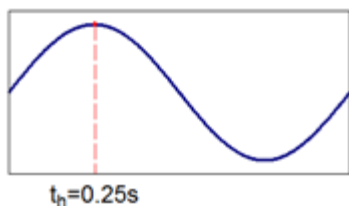
You have two options available for customizing the coordinates window. Instead of displaying the predefined coordinates, you can display any presentation object in the coordinates window, such as a table with custom results related to the cursors. FlexPro updates this object automatically whenever cursors are moved. The second option is to extend the coordinates window to include custom coordinates. These are special FPScript formulas that access the cursors and perform particular calculations, such as finding the area under the curve between the two cursors. FlexPro provides you with several predefined coordinates. However, you can also program your own coordinates in FPScript.

### Dimensioning curves

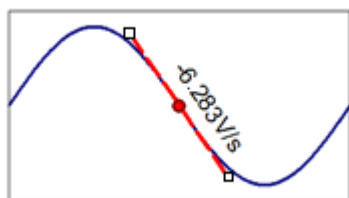
FlexPro offers you a variety of dimensioning elements that you can use to dimension the curves of diagrams:

**Dimensioning**

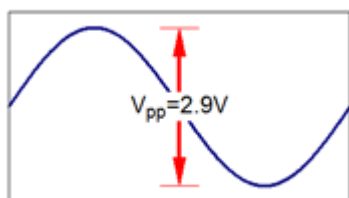
Boundary Line



Perpendicular Line



Tangent



Dimension Line

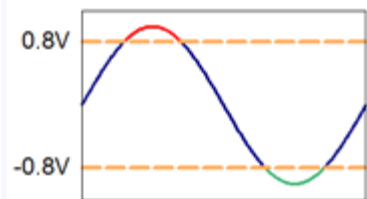
**Description**

A horizontal or vertical boundary line.

A perpendicular line from a particular point on the curve to one of the edges of the diagram.

A tangent at a point on the curve.

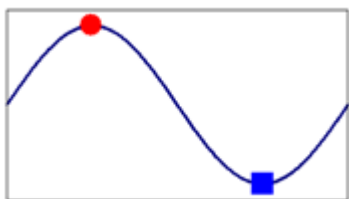
A dimensioned distance in the Y direction, X direction or in any direction.

**Dimensioning**

Tolerance Band

**Description**

A horizontal or vertical tolerance band. A color change can be enabled for the curve when the tolerance is overshoot or undershot.



Symbol

A symbol that highlights a particular point on the curve.

Dimensioning is added as additional curves in the diagram so that they are always correctly scaled with the diagram when subsequently zooming into the curve.

The current cursor positions are used as the initial points for the coordinates used for new dimensioning. These must not necessarily be on the curve. You can work with free cursors.

**Accessing Cursors and Markers with FScript**

You can access the current cursor positions and markers with FScript and thus use these within analyses. For instance, using both cursors you can select a range for which an analysis is to be calculated, or you can select events in a signal using markers that you can then access in FScript formulas. To do this, the diagram, document and the worksheet provide you with Properties with which you can access the cursors and markers. For example, the following FScript formula accesses the current cursor positions of the second pane of a worksheet in order to form a signal segment:

```
Signal[Worksheet.WKS.Cursor(fpCursorMinX, 2).PositionIndex, Worksheet.WKS.Cursor(fpCursorMaxX, 2).PositionIndex]
```

## 4.1 Toggling Cursors

### Toggling Cursors in a Document or Diagram

- Use the command **Toggle Cursors** from the **Cursors** menu.

---

**Note:** You cannot disable cursors in a worksheet.

---

### Toggling Harmonic Cursors

- Select or deselect the option **Harmonic Cursors** in the **Cursors** menu or on the **Cursors** toolbar.

### Changing the Number and Spacing of Harmonic Cursors

1. Right-click with your mouse on the diagram, worksheet or document for which you would like to change the display.
2. In the context menu that appears, select the **Properties** command to open the Properties dialog box.
3. Select the **Cursors** tab.
4. Enter the desired **number** and **spacing** of cursors in octaves.

### Excluding a Curve from Cursorsing

#### The cursors enabled:

1. Use the right mouse button to click on the name of the curve in the coordinates window that you would like to exclude.
2. In the context menu, choose **Exclude Curve**.

#### With cursors disabled:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its label that is to be disregarded when cursoring.
3. In the dialog box that appears, click on the **General** tab.

4. Uncheck the option **Account for this curve when using cursors**.

## 4.2 Moving the Cursors

### Moving the Cursors Using the Keyboard

Move the cursors in the X direction using the RIGHT and LEFT keys, in the Y direction using the UP and DOWN keys and in the Z direction using the PAGE UP and PAGE DOWN keys. If you do this while holding down the CTRL key, the increment will be increased slightly, and the increment will be increased even more if you use the ALT key.

All commands for navigating cursors are available in the **Navigation** submenu of the **Cursors** toolbar, or in the **Cursors** menu. Point with your mouse at one of the commands and the key assignment will be displayed.

On the **Cursors** toolbar, select the option **Synchronize cursors** if the diagram is located in a worksheet or a document and the cursors of the other diagrams are also to be moved synchronously.

The following table shows additional key assignments:

Key	Function
TAB	Switches between origin cursor and leading cursor.
F6	Switches to the next diagram in the document or worksheet.
SHIFT+F6	Switches to the previous diagram in the document or worksheet.

### Moving the Cursors with the Mouse

In principle, you move free and bound cursors in the same way. However, the bound cursor's movement is limited in that it must always follow the curve.

Movement	Procedure
Move one cross-hair cursor or a line cursor horizontally	Use your mouse to drag the vertical cursor line to the desired position.
Move both cross-hair cursors or line cursors horizontally	While holding down the ALT key, use your mouse to drag the vertical cursor line to the desired position.
Move one cross-hair cursor vertically	Use your mouse to drag the horizontal cursor line to the desired position.



Movement	Procedure
Move both cross-hair cursors vertically	While holding down the ALT key, use your mouse to drag the horizontal cursor line to the desired position.
Move one cross-hair cursor freely	Use the mouse to drag the point where the two cursor lines cross to the desired position.
Move both cross-hair cursors or line cursors freely	While holding down the ALT key, drag the rectangle enclosed by the cross-hairs or the vertical band limited by the line cursor in the desired direction.
Place a cursor onto a specific point on the curve.	Use your mouse to click on the desired point. The mouse cursor must appear as an arrow. If this is not the case, for example, because the point is located between the cursors or under a cursor line, then you have to hold down the SHIFT key.

---

**Note:** If you would like to place a bound cursor onto a specific point on the curve, you only have to click on this point. FlexPro must then search for the point on the curve closest to the position clicked. Here you can choose between the options **To nearest point** and **Via X position**. If the first option mentioned is active, then FlexPro searches through the entire curve to determine the point that is closest to the position clicked. In this mode, it is very easy to select a peak value, for example. However, for larger data sets, the method is very computing-intensive, which is why you should then activate the other method. Only the horizontal component of the position clicked is taken into account and from this, the index of the point to be selected is directly calculated. This method is very fast because you don't have to search through the data set; however, it only works with equidistantly sampled signals. You can select the options on the **Cursors** tab of the Properties dialog box for the diagram, worksheet or document whose cursor you want to use.

Select the option **Synchronize cursors** on the **Cursors** toolbar if the cursors of the other diagrams are to be moved synchronously within a document or worksheet.

---

## Navigating to Peak Values or Markers

You can use the commands **Cursor on Previous Event** and **Cursor on Next Event** from the **Navigation** submenu of the **Cursors** toolbar to move the active cursor from peak value to peak value or from marker to marker. On the **Cursors** tab of the diagram's Properties dialog box, you can specify whether you want to go to local maxima, local minima, minima and maxima or markers.

Here you can also enter a hysteresis, which determines what should or should not be recognized as a peak value. If some are skipped when navigating the peak values, you need to decrease the hysteresis. If small waves or noise in the course of the signal are recognized as peak values, then you need to increase the hysteresis.

### Navigating to a Specific Point in a Diagram

Use the **Go To** command from the **Navigation** submenu on the **Cursors** toolbar or the **Cursors** menu to place the cursor specifically on a point of the curve or a point in the diagram plane. If the option **Keep Cursor Visible** is enabled, the image section is automatically adapted, if necessary.

You can also navigate to a point using your mouse by simply clicking on the point.

### Navigating to a Frame in a Video

Use the **Go To** command from the **Navigation** submenu on the **Cursors** toolbar or the **Cursors** menu to navigate to a frame in a video. Alternatively, you can use the horizontal scrollbar.

### Scrolling along a Curve

If you display the curve spread out in the diagram as follows and then animate the cursor, you will obtain a scrolling effect:

1. Hold down the CTRL key and use your mouse to drag a vertical band from left to right in order to display a section of the curve as spread out.
2. In the **Cursors** menu, point to **Adjust Image Section** and then in the submenu select the option **Fixed Cursor Position**.
3. Uncheck the option **Cross-Hair Cursor** on the **Cursors** toolbar in order to display the cursor as a **line cursor**.
4. Now click on the **Playback** toolbar's **Playback** or **Reverse Playback** button to allow the cursor move to the right or left.

---

#### Notes:

- The **Fixed Cursor Position** option causes the cursor to appear at a fixed location and the curve to move under the cursor. Alternatively, you can use the **Move Image Section** option.
- You can change the speed of the cursor in the **Playback Speed** box on the

**Playback** toolbar.

- You can set the cursoring increment in the **Playback increment** box on the **Cursors** tab of the Properties dialog box.
- 

## Synchronizing Cursors

To synchronize the cursors of several diagrams in a worksheet or document:

1. Click with your right mouse button on the worksheet or document and choose **Properties** from the context menu.
2. Switch to the **Cursors** tab.
3. Under **Synchronization**, select the option **Synchronize cursors**.
4. If the data sets based on the curves have different sampling rates, select the option **via X value**.

## 4.3 Changing the Image Section

### Stretching the Image Section

1. Place the two cursors on the left and right side of the section that you want to display as stretched.
2. Choose the **Spread X Range Between Cursors** command on the **Cursors** toolbar. You can run the command from the keyboard by using the PLUS key (+) on the numeric keypad.

---

**Note:** Spreading takes place in such a way that both cursors end up on the left and the right side of the diagram.

---

### Synchronizing Zoom Actions

1. Click on the diagram in which you want to synchronize zooming actions with the zooming actions of another diagram.
2. Right-click with your mouse on the worksheet or document and choose **Properties**.
3. Switch to the **Cursors** tab.
4. Select the option **Zoom in sync**.
5. Specify in the **Synchronization mode** field if you want to synchronize via the **X**

**value** or the **data index**.

6. If zoom operations are to be synchronized in the Y direction as well, select the option **Also zoom sync. in Y direction**.

## Stretching and Compressing the Curve

- Use the commands **Stretch Curve Vertically**, **Compress Curve Vertically**, **Stretch Curve Horizontally** or **Compress Curve Horizontally** from the submenu **Change Image Section** on the **Cursors** toolbar or in the **Cursors** menu.

---

**Note:** For the axis concerned, automatic scaling is turned off and the axis end values are reduced or increased by 10% of the range of values displayed.

---

## Using the Mouse to Zoom

### To display any rectangular section as a close-up:

- Use your mouse to drag a rectangle from the top left to the bottom right of the desired section to enclose it.

### To enlarge the graphic in the X direction:

- Press and hold the CTRL key and use your mouse to drag a vertical band from left to right.

### To enlarge the graphic in the Y direction:

- Press and hold the SHIFT key and use your mouse to drag a band from top to bottom.

## Scrolling

- If the active curve is not completely visible, you can scroll through the section of the image displayed in the diagram. To do this, you can use the scrollbars displayed at the bottom and to the right of the worksheet or the active diagram in the document.
- You can use the commands from the submenu **Navigation** on the toolbar or in the **Cursors** menu to scroll at a fixed increment.
- You can use the mouse wheel to scroll instead of the scrollbars. You can scroll vertically by holding down the CTRL key while using the mouse wheel. Otherwise,

scrolling takes place horizontally.

### Adapting the Image Section to the Curve

- Select one of the commands **Curve At Full Height** and **Show At Full Width** or **Display Complete Curve** from the submenu **Change Image Section** of the **Cursors** menu or on the **Cursors** toolbar.

---

**Note:** When using the command for the first time, the corresponding axis of the diagram is switched to autoscaling via the data range lower limit/upper limit. If the data range limits are specified in the data set, they are then used as the default for autoscaling the axis. If the command is executed again, it switches to autoscaling of the axis via the data minimum/maximum.

---

### Adjusting the Image Section when Moving Cursors

Use the **Adjust Image Section** option on the toolbar or in the **Cursors** menu to determine how the image section should be adjusted when you move a cursor:

- **Do Not Adjust Image Section** leaves the segment unchanged. Use this option if you do not want to change the configured axis scaling.
- **Fixed Cursor Position** adjusts the image section so that the cursor appears in a fixed location. For a cross-hair cursor the adjustment is in the X and Y direction, and for a line cursor it is only in the X direction.
- **Expand Image Section** automatically expands the image section when the cursor leaves.
- **Move Image Section** automatically moves the image section when the cursor leaves.

### Undoing Zoom Actions

- You can undo all actions in increments that change the section displayed in the diagram using the command **Restore Previous Image Section** from the **Cursor** menu or toolbar.
- Use the command **Restore Original Image Section** to restore the axes of the diagram to their original condition.

---

**Note:** You can run the command from the keyboard by using the MINUS key (-) on the numeric keypad.

---

## Automatic Zooming of Segments

You can zoom closer to the area between the cursors of a diagram in a second diagram. To do this, proceed as follows:

1. Select the diagram in which you want get a close-up of a segment.
2. Select the option **Automatic Zoom** on the **Cursors** toolbar to enable automatic zooming for this diagram.
3. Now switch to a different diagram and move the cursors.

## 4.4 Working with Markers

### Setting and Deleting Markers

#### To set a marker for display in all diagram views:

1. Set the cursor to the point on the curve that you want to mark.
2. Choose the **Set Marker** command on the toolbar or in the **Cursors** menu or press the ENTER key.

#### To set a marker for display only in the active diagram view:

1. Set the cursor to the point on the curve that you want to mark.
2. Hold down the SHIFT key and press the ENTER key.

#### To set a range marker for display in all diagram views:

1. Set the two cursors to the right and left of the range of the curve you want to mark.
2. Choose the **Set Range Marker** command on the toolbar or in the **Cursors** menu or press the ENTER key while holding down the CTRL key. In the plan view of a 3D-surface diagram, you can select a rectangular range by using your mouse while holding down the CTRL key and SHIFT key to draw a rectangle around the range.

**To set a range marker for display only in the active diagram view:**

1. Set the two cursors to the right and left of the range of the curve you want to mark.
2. Press the ENTER key while holding down the SHIFT and CTRL keys.

**To set an image marker for display in all diagram views:**

1. Set the cursor to the point on the curve that you want to mark.
2. Select **Set Image Marker** command on the toolbar or in the **Cursors** menu or press the ENTER key while holding down the ALT key.
3. If you specified a media on the diagram's **Cursors** tab, a still image will be used from that media. Otherwise, a dialog box appears where you can select an image file.

**To set an image marker for display only in the active diagram view:**

1. Set the cursor to the point on the curve that you want to mark.
2. Press the ENTER key while holding down the SHIFT and ALT keys.
3. If you specified a media on the diagram's **Cursors** tab, a still image will be used from that media. Otherwise, a dialog box appears where you can select an image file.

**To delete a marker:**

- Delete the marker closest to the active cursor using the command **Delete Marker**.
- You can delete all of a curve's markers by placing a cursor on the curve and then using the **Delete All Markers** command.
- You can delete all unbound markers by disabling the option **Bind Cursor To Curve** and then using the **Delete All Markers** command.
- You can delete any marker while the cursors are active by clicking on it with your right mouse button and selecting **Delete Marker** from the context menu.
- If you disable the cursors by clicking on **Toggle Cursors**, you can select one or more markers and delete them using the DEL key. If the diagram is in a document or worksheet, you will first have to open it by double-clicking on it.

## Positioning Markers

1. Click on the marker that you would like to position and keep the mouse key pressed.
2. Now drag the marker onto the desired position.

---

**Note:** When you position a marker for display in all diagram views, then it appears in all views at the new position.

---

## Assigning Media as a Source for Image Markers

### When cursors are enabled:

1. Right-click with your mouse on a marker and select **Marker Properties**.
2. On the **Markers** tab of the Properties dialog box that appears, select **File** from the **Source** field.
3. Under **Media**, enter the media whose frames should serve as the source for the image markers in the diagram.

### In editing mode:

1. If the diagram is in a document, select the diagram.
2. Now right-click with your mouse on the diagram and select **Properties**.
3. On the **Markers** tab of the Properties dialog box that appears, select **File** from the **Source** field.
4. Under **Media** enter the media whose frames should serve as the source for the image markers in the diagram.

## Editing Marker Text

### To edit the default text to be displayed in all markers to which you did not assign an individual text:

#### When cursors are enabled:

1. Right-click with your mouse on a marker and select **Marker Properties**.
2. On the **Markers** tab of the Properties dialog box that appears, select **File** from the **Source** field.



3. Under **Media**, enter the media containing the still images that will serve as source for the image markers in the diagram.

**In editing mode:**

1. If the diagram is in a document, select the diagram.
2. Now right-click with your mouse on the diagram and select **Properties**.
3. Enter the desired text on the **Markers** tab of the Properties dialog box that appears. You can embed fields for the index, X, Y and Z values of the marker, and for the corresponding units.

**To assign individual text to a marker:**

**When cursors are enabled:**

1. Right-click with your mouse on the marker and select **Edit Marker**.
2. In the **Edit Marker** dialog box, you can assign text to a marker or edit existing text.

**In editing mode:**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a marker to select it.
3. Press the F2 key.
4. In the **Edit Marker** dialog box, you can assign text to a marker or edit existing text.

**To reapply the default text to a marker that displays unique text:**

**When cursors are enabled:**

1. Right-click with your mouse on the marker and select **Edit Marker**.
2. Delete the existing text in the **Edit Marker** dialog box.

**In editing mode:**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a marker to select it.
3. Press the F2 key.
4. Delete the existing text in the **Edit Marker** dialog box.

**Note:** You can also assign text to a range marker, but this will not be displayed in the curve.

---

## Changing Marker Size

**To change the size of all markers to which you have not assigned any individual size attribute:**

**When cursors are enabled:**

1. Right-click with your mouse on a marker and select **Marker Properties**.
2. On the **Markers** tab of the Properties dialog box that appears, enter the desired **Width** and **Height**. For text markers, these are the minimum values to be exceeded if the text is larger.

**In editing mode:**

1. If the diagram is in a document, select the diagram.
2. Now right-click with your mouse on the diagram and select **Properties**.
3. On the **Markers** tab of the Properties dialog box that appears, enter the desired **Width** and **Height**. For text markers, these are the minimum values to be exceeded if the text is larger.

**To change the size of a single marker:**

**When cursors are enabled:**

1. Right-click with your mouse on the marker and select **Edit Marker**.
2. In the **Edit Marker** dialog box, enter the desired **Width** and **Height**. For a text marker, these are the minimum values to be exceeded if the text is larger.

**In editing mode:**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a marker to select it.
3. Press the F2 key.
4. In the **Edit Marker** dialog box, enter the desired **Width** and **Height**. For a text marker, these are the minimum values to be exceeded if the text is larger.

**To reassign the default size to a marker with an adjusted size:****When cursors are enabled:**

1. Right-click with your mouse on the marker and select **Edit Marker**.
2. In the **Edit Marker** dialog box, enter 0 in the **Width** and/or **Height** field.

**In editing mode:**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a marker to select it.
3. Press the F2 key.
4. In the **Edit Marker** dialog box, enter 0 in the **Width** and/or **Height** field.

---

**Note:** You can also assign text to a range marker, but this will not be displayed in the curve.

---

## Changing Marker Formatting

**To change the formatting for all markers to which you have not assigned any individual text:****When cursors are enabled:**

1. Right-click with your mouse on a marker and select **Marker Properties**.
2. On the **Markers** tab of the Properties dialog box that appears, set the insertion point in the **Text** input field on the placeholder of the value whose formatting you want to change, such as **%<YValue>** for the Y value.
3. Now click on **Edit Formatter** and set the formatting desired.

**In editing mode:**

1. If the diagram is in a document, select the diagram.
2. Now use your right mouse button to click on the diagram and select **Properties**.
3. On the **Markers** tab of the Properties dialog box that appears, set the insertion point in the **Text** input field on the placeholder of the value whose formatting you want to change, such as **%<YValue>** for the Y value.
4. Now click on **Edit Formatter** and set the formatting desired.

### To change the formatting for a single marker:

#### When cursors are enabled:

1. Use your right mouse button to click on the marker and select **Edit Marker**.
2. In the **Edit Marker** dialog box that appears, set the insertion point in the **Text** input field on the placeholder of the value whose formatting you want to change, such as **%<YValue>** for the Y value.
4. Now click on **Edit Formatter** and set the formatting desired.

#### In editing mode:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a marker to select it.
3. Press the F2 key.
4. In the **Edit Marker** dialog box that appears, set the insertion point in the **Text** input field on the placeholder of the value whose formatting you want to change, such as **%<YValue>** for the Y value.
5. Now click on **Edit Formatter** and set the formatting desired.

### Customizing the Display of Markers

#### To change the display attributes of all markers to which you have not assigned any individual display attributes:

#### When cursors are enabled:

1. Right-click with your mouse on a marker and select **Marker Properties**.
2. On the **Markers** tab of the Properties dialog box that appears, set the attributes for **Marker label display**, **Line display**, **Colors** and **Font**.

#### In editing mode:

1. If the diagram is in a document, select the diagram.
2. Now right-click with your mouse on the diagram and select **Properties**.
3. On the **Markers** tab of the Properties dialog box that appears, set the attributes for **Marker field label display**, **Line display**, **Colors** and **Font**.

**Note:** You fill the markers automatically with the color of the particular curve by selecting **Automatic** as the background color.

---

### **To assign individual display attributes to a marker:**

#### **When cursors are enabled:**

1. Right-click with your mouse on the marker and select **Edit Marker**.
2. In the **Edit Marker** dialog box, set the colors for **Text**, **Background**, **Border** and **Line**.

#### **In editing mode:**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a marker to select it.
3. Press the F2 key.
4. In the **Edit Marker** dialog box, set the colors for **Text**, **Background**, **Border** and **Line**.

### **To restore default attributes assigned to a marker that uses individual display attributes:**

#### **When cursors are enabled:**

1. Right-click with your mouse on the marker and select **Edit Marker**.
2. In the **Edit Marker** dialog box, select in the menu for the corresponding color **No Text**, **No Fill** or **No Line**.

#### **In editing mode:**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a marker to select it.
3. Press the F2 key.
4. In the **Edit Marker** dialog box, select in the menu for the corresponding color **No Text**, **No Fill** or **No Line**.

## Copying Markers to the Clipboard

1. If applicable, toggle the cursors using the command **Toggle Cursors** from the **Cursors** menu.
2. Choose the command **Copy Markers** from the Cursors menu.
3. In the dialog box that appears, choose whether you would like to save the indices of the markers or the values of the markers. The **indices of the markers** are saved as a data series, which contains the indices of the selected points in the order in which you marked them. The **values of the markers** are saved as a signal that contains the X and Y components of the points selected.
4. To save the data set in the Clipboard to the open project database folder, click with your right mouse button on the Object List and select **Paste** from the context menu.

## 4.5 Working with Data

### Editing Data

#### To change the Y value under the active cursor:

1. Set the cursor to the point on the curve whose Y value you want to mark.
2. Open the **Cursors** menu, point to **Change Value(s)** and select the command **Edit Value Under Cursor**. You can also press the INSERT key instead.
3. In the dialog box that appears, enter a new Y value in the physical unit of the data upon which the curve is based.

#### To change the Y value under the active cursor using the mouse:

1. Hold down the CTRL and ALT keys and move the mouse pointer to the location where the vertical and horizontal cursor lines cross. The mouse pointer changes to a double arrow with an x in the middle.
2. Now hold down the left mouse button and drag the point under the cursor to the desired Y position.

#### To change the Y values of all data between the cursors:

1. Set both cursors on the left and right edge of the area whose Y values you want to change.
2. Open the **Cursors** menu, point to **Change Value(s)** and select the command **Set Values Range Between Cursors To New Value**.

3. In the dialog box that appears, enter a new Y value in the physical unit of the data upon which the curve is based.

---

**Note:** You can only manipulate data directly if the curve is based on a data set and not on a formula. If the curve is based on a formula, you first have to convert it into a data set.

**Changes are made directly in the data set and cannot be undone.**

---

## Setting Data to Void

### To make the Y value under the active cursor void:

1. Set the cursor to the point on the curve whose Y value you want to make void.
2. Open the **Cursors** menu, point to **Change Value(s)** and select the command **Make Value Under Cursor Void**.
3. In the dialog box that appears, enter a new Y value in the physical unit of the data upon which the curve is based.

### To make the Y values of all data between the cursors void:

1. Set both cursors on the left and right edge of the area whose Y values you want to make void.
2. Open the **Cursors** menu, point to **Change Value(s)** and select the command **Make Values in Range Between Cursors Void**.
3. In the dialog box that appears, enter a new Y value in the physical unit of the data upon which the curve is based.

## Interpolating Data

### To interpolate the Y value under the active cursor:

1. Set the cursor to the point on the curve whose Y value you want to interpolate.
2. Open the **Cursors** menu, point to **Change Value(s)** and select the command **Interpolate Value Under Cursor**.
3. In the dialog box that appears, enter a new Y value in the physical unit of the data upon which the curve is based.

**To interpolate the Y values of all data between the cursors:**

1. Set both cursors on the left and right edge of the area whose Y values you want to interpolate.
2. Open the **Cursors** menu, point to **Change Value(s)** and select the command **Interpolate Values in Range Between Cursors**.
3. In the dialog box that appears, enter a new Y value in the physical unit of the data upon which the curve is based.

**Copying Data to the Clipboard**

**To copy the Y values to the Clipboard that are currently displayed in the coordinates window for the active cursor as a data series:**

- Use the command **Copy Value(s)** on the **Cursors** toolbar or in the **Cursors** menu.

**To copy the section of the curve between the cursors to the Clipboard:**

1. Use the command **Copy Range** on the **Cursors** toolbar or in the **Cursors** menu.
2. In the dialog box that appears, specify whether the current cursor positions are to be inserted as fixed or variable values. The command saves a formula in the clipboard, which extracts the section from the curve. In the latter case, the properties of the diagram, worksheet or document are used that determine the positions of the cursors. Thus, for example, you can combine the manual selection of a signal section with an automated analysis.

**Displaying Data of a Curve**

1. Set the cursor on the curve whose data you want to display.
2. Choose the command **Open Data Set** on the **Cursors** toolbar.

## **4.6 Working with the Coordinates Window**

**Displaying and Hiding the Coordinates Window**

- To display or hide the coordinates window, click on **Toggle Coordinates Window** on the **Cursors** toolbar.



## Changing Coordinates Window Display Options

1. Use your right mouse button to click on the coordinates window and select **Coordinates Properties**.
2. On the **Coordinates Window** tab, select the desired options under **Display in window**.

## Changing the Coordinates Window Output Format

You can change the output format in the coordinates window for the data of one or more curves. The formatter that determines the output format is stored in the diagram on the **General** tab of the curve.

1. Use the right mouse button to click on the value in the coordinates window for which you would like to change the output format.
2. Choose the command **Change Output Format**.
3. Set the desired formatting in the **Formatter Properties** dialog box and click on OK to close the dialog box.
4. If several curves are to be displayed in the coordinates window, the **Assign Formatting** dialog box appears. Here you can specify whether the new formatting is also to be used for the additional curves.

## Opening a Data Set Displayed in the Coordinates Window

- Double-click in the coordinates window on the name of the curve based on the data set or right-click with the mouse on the name and select **Open Data Set** from the context menu.

## Copying a Value from the Coordinates Window

### To copy a value to the clipboard:

1. Use the right mouse button to click on the value in the coordinates window.
2. From the context menu, choose the **Copy** command to copy a value to the clipboard.

### To copy a value using drag-and-drop:

1. Use the left mouse button to click on the value in the coordinates window.

2. While holding down the mouse button, drag the value to the desired target object and then release the button. If the window of the target object is not visible, move the mouse to the object's tab and wait until the window appears in the foreground.

---

**Note:** You can copy the value to the Object List, to the data view or to any presentation object, such as a table or text object.

---

## Copying the Contents of the Coordinates Window to the Clipboard

You can copy the content of the coordinates window to the clipboard as a picture and insert it into a diagram, a table or a document. This is a static copy, which means that the figures in the copy will not change if you move the cursor at a later stage.

1. Use the right mouse button to click on the coordinates window.
2. From the context menu, choose the **Copy Coordinates** command to copy the content to the clipboard.
3. To insert it in the target object, open it and select the **Paste** command from the **Edit** menu.

## Inserting Dynamic Coordinates into a Diagram, Document or Worksheet

You can insert a dynamic copy of the content of the coordinates window into a diagram, document or worksheet. If you should move the cursor at a later stage, the numbers in the copy will change accordingly.

1. From the menu bar, open the **Document, Diagram or Worksheet** menu.
2. Choose the command **Insert Coordinates**.
3. Place the copy using the mouse.

---

**Note:** If you disable the cursors, the inserted coordinates will be frozen and the content last displayed will remain.

---

## Adding a Custom Coordinate

1. Use your right mouse button to click on the coordinates window and select **Add Coordinate**.
2. In the **Select Coordinate** dialog box, choose one of the coordinates provided from the list or choose (**New custom coordinate**) to program a coordinate in FScript.
3. To program your own coordinate, follow the instructions displayed in the **Custom Coordinate Properties** dialog box.

## Editing a Custom Coordinate

1. Click with your right mouse button in the coordinates window on the coordinate to be edited and choose **Edit Coordinate**.
2. Edit the coordinate in the **Custom Coordinate Properties** dialog box.

---

**Note:** Text the FScript code with the Debugger before closing the dialog box.

---

## Displaying and Hiding a Custom Coordinate

### To hide a coordinate:

- Click with your right mouse button in the coordinates window on the coordinate you want to hide and choose **Hide Coordinate**.

### To delete a coordinate permanently:

1. Use your right mouse button to click on the coordinates window and select **Coordinates Properties**.
2. On the **Coordinates Window** tab, select the coordinate to delete from the **Custom Coordinates** list.
3. Click on **Remove Coordinate**.

### To display the hidden coordinate again:

1. Use your right mouse button to click on the coordinates window and select **Coordinates Properties**.
2. On the **Coordinates Window** tab, select the coordinate that you want to show from the **Show and Name** column of the **Custom Coordinates** list.

## Displaying a Presentation Object in the Coordinates Window

In the coordinates window, you can display any presentation object, such as a table with embedded calculation results. The text is then automatically updated each time the cursor is moved, so that embedded calculations referring to the current cursor positions always remain current.

### Using the Properties dialog box:

1. Double-click on a free space in the coordinates window.
2. On the **Coordinates Window** tab, select the option **Object** under **Display in window**.
3. In the text box, specify the name of the object to be displayed with the extension corresponding to the object type, or select an object from the list.

### Using drag-and-drop:

1. Enable the cursors in the object in whose coordinates window you would like to display a presentation object so that the coordinates window appears.
2. Now drag the object that should appear in the coordinates window from the Object List into the coordinates window. Make sure that you do not activate the Object List by clicking on it, otherwise the coordinates window will disappear.

---

### Note:

- The name of the text is displayed in the title bar of the coordinate window.
  - The size of the text determines the size of the coordinates window. To reduce the size of the object, you have to open it and then reduce the size.
- 

### Example

On your hard disk you can find the **Examples** project database. In the **Cursor** folder of the database, you can find an example of how the coordinates window can be used to display analysis results.

The project database path name under Windows Vista or Windows 7 is usually C:\Users\Public\Documents\Weisang\FlexPro\9.0\Examples\Examples.fpd or C:>Public>Public Documents>Weisang>FlexPro>9.0>Examples>Examples.fpd and under Windows 2000 or XP it is C:\Documents and Settings\All Users\Shared Documents\Weisang\FlexPro\9.0\Examples\Examples.fpd. If you cannot find the project database on your computer, you may have to install the database. To do this, use the **Custom** setup type and select **Example databases** for the installation.

## Dimensioning Curves

1. Enable the cursors of the diagram whose curve you would like to dimension or of the document where the diagram is located.
2. If the diagram is in a document or worksheet, click on it to select it.
3. Position the cursors on the curve to be dimensioned.
4. Select **Dimension Curve** on the **Cursors** toolbar.
5. In the dialog box that appears, select the tab of the dimension that you want to add.
6. The X and Y values are pre-filled with the current cursor positions. If necessary, you can change the coordinates. You have to specify the values in the physical unit of the respective curve component.
7. If applicable, set additional attributes for the selected dimensioning, such as the direction of a boundary line, on the tab.
8. Predefine the line attributes for the dimensioning under Line.
9. Under Labeling, specify a text, if necessary, with which the dimensioning is to be labeled. Choose among the fields provided in the Fields list in order to add coordinates, distance, slope, etc.
10. Specify in the **Location** field the location, if applicable, of the label on the dimension line and its position relative to the selected location in the **Position** field.

## Translating Curves

### To translate a curve in increments using the keyboard:

1. Set the cursor on the curve you want to translate.
2. If necessary, press the NUM LOCK key on the numeric keypad of your keyboard in order to turn off NUM LOCK mode.
3. Use the arrow keys on the numeric keypad correspond to the numbers 2, 4, 8 and 6 in order to translate the curve in the particular direction. You can increase the increments by holding down the CTRL or ALT key. The corresponding commands are in the **Translate Curve** menu on the **Cursors** toolbar, or in the **Cursors** menu.

### To translate a curve using the mouse:

1. Hold the CTRL key down and move the mouse to the horizontal or vertical line of the active cursor or to the location where the two lines cross, depending on whether you want to translate the curve vertically, horizontally or in both directions.

2. Now hold down the left mouse button and drag the curve with the mouse to the desired position.

### **To translate a curve around a predefined numeric value.**

1. Disable the cursors using the command **Toggle Cursors** from the **Cursors** menu.
2. Click on the curve or on its axis label to select it.
3. In the Properties window, enter the desired values in the fields **X offset** and **Y offset** in the unit of the data upon which the curve is based.

### **To align two curves:**

1. Place the original cursor on the curve that is not to be translated, but is to serve as a reference for the translation.
2. Move the cursor to the desired reference point on this curve.
3. Now activate the leading cursor and place it on the curve to be translated.
4. Move the cursor to the point on the curve to be aligned with the reference point of the other curve.
5. Select one of the commands **Translate Curve Horizontally and/or Vertically So Both Cursors Lie On Top Of Each Other**. The commands are in the **Translate Curve** menu on the **Cursors** toolbar or in the **Cursors** menu.

---

**Note:** The data of the translated curve is not changed during translation. The curve just appears as translated. The original data of the data set upon which the data is based continues to be displayed in the coordinates window as well. By translating the curve, its data deviates from the axis scaling.

---

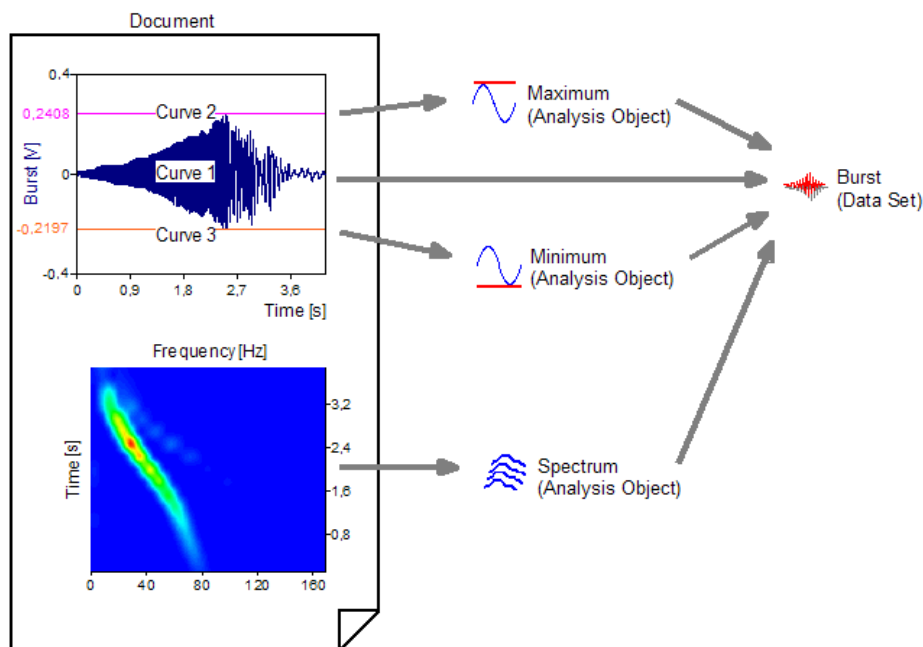
### **Canceling a Curve Translation**

1. Place the cursor on the curve whose translation you want to undo.
2. Select one of the commands **Reset Horizontal and/or Vertical Translation of Curve**. The commands are in the **Translate Curve** menu on the **Cursors** toolbar or in the **Cursors** menu.

## 5 Analyzing Data Mathematically

Traditional applications for data analysis use a number of analysis commands to generate new data sets from the raw data sets available, which then contain the results of the analysis. In FlexPro, however, you set up a dynamic network of analysis objects and formulas to evaluate your data. Each analysis object and each formula represent the result of an analysis, such as the mean value or the frequency spectrum of a signal. Formulas are programmed in the FlexPro FScript analysis language. Analysis objects are special formulas that are managed by FlexPro and for which you can set parameters via the Properties dialog box. To calculate an analysis result, the FScript code of the formula or the analysis object is processed to calculate the result. When processing the formula, FlexPro accesses the source data on which this formula is based and to which the formula is linked. For an analysis object, specify the source data to be used on the **Data** tab of the Properties dialog box. For a formula, use the source data as you would a variable in FScript code. You can nest the object network that you are creating as deeply as you like. Not only can the source data of an analysis object or a formula be data sets, but the data can also be different analysis objects or formulas. For all objects that can supply data, i.e. for formulas, analysis objects and data link objects, FlexPro uses the term data object. You can use these types of data objects like data sets and display them as a diagram or table, for instance.

The following illustrates a simple analysis network:



The object-oriented data analysis from FlexPro has two distinct advantages over the conventional approach. First, you can modify the analysis objects and formulas at any time to optimize the analysis; second, you have the option of using the analysis that was created just once for as many new data sets as you would like without having to repeat all of the steps in the analysis. All settings that you make during the analysis remain stored in FlexPro's network of data objects. After changing a setting or after importing new data, you only need to use the Update command to update all documents with the latest information.

If you are new to FlexPro and used the conventional methods for data analysis in the past, the object-oriented application from FlexPro will require some changes in your thought process. However, you will learn to appreciate the advantages of this method rather quickly.

## Analysis Wizard

The Analysis Wizard offers you a wide range of ready-made analyses to choose from and creates a complete evaluation of the data sets selected by you. The final analysis does not only contain the desired calculation results, but as an option also contains specially



adapted diagrams and tables that present these results. If required, the wizard can even create a finished document for your analysis.

### **Analysis Templates**

You can store analyses created one time as an analysis template in the current database or in a template database. This type of analysis template consists of any network of interdependent objects. In the example above, these would be the document, the diagram within it, and three analysis objects. Objects in the lowest hierarchical level, i.e. those that do not depend on any other objects, are replaced by selected objects during subsequent use of the template, as long as they were not marked as belonging to the template when it was created. In the example above, this would be the data set. However, this can also be any other type of object, such as a diagram. In the Analysis Wizard you can then use the analysis templates to create new analyses.

### **Analysis Automation**

All analyses in FlexPro are created using analysis objects or formulas, allowing for automation of the analysis without any programming.

To update a document or worksheet, you only need to run the **Update** command. FlexPro then updates all diagrams, text and tables in that object. If, for example, a formula or an analysis object has been entered as a curve, FlexPro checks to see whether a new calculation is required and, if necessary, carries out the re-calculation. If, in turn, the formula uses other formulas and/or data sets in calculations, then this network is completely processed.

If you have set up data link objects for importing external data, the original data is accessed with each update, as long as it has changed since last updated. Changes thus affect the results immediately.

### **Smart Update**

When you update a document or diagram, all of the formulas and analysis objects upon which this document is based are also updated. This type of network can quickly cover several hundred formulas. FlexPro "Smart Update" technology ensures that only the formulas with indeterminate results are re-calculated. If a formula was already calculated during a previous updating process, then the last result is re-used, if this is at all possible.

FlexPro checks the formula code here for vagueness, which means that re-calculation could provide a different result. Only those formulas are re-calculated for which this is the case. A typical example of a formula with a vague result is a formula that uses the Noise function or the Input function. Formulas that refer to data sets or results of other

formulas are re-calculated if at least one of the data sets has been edited or if one of the formulas used has to be re-calculated.

FlexPro also ensures that each formula is not calculated more than once within a single update. You can therefore assume that a random signal used in several formulas represents the same instance in each of these formulas. An exception to this are formulas with arguments. These are always calculated.

However, "Smart Update" includes not only formulas, but presentation objects as well. For a diagram, FlexPro checks for each individual curve, for instance, to see whether it must be re-plotted.

## 5.1 Working with Analyses

### Structuring Data Analyses

The most important task when analyzing data using FlexPro is to structure the analysis correctly, which means that the analysis problem should ideally be broken down into a network of analysis objects and formulas. Provided that you are working exclusively with the analysis objects provided by FlexPro, this structure is automatic. Just select the data set to be analyzed and choose an analysis object to obtain the relevant result. For instance, if you want to display the signal minimum and maximum in your document, create two analysis objects **Statistical Quantity**, one for the maximum and one for the minimum.

If, however, you use your own algorithms, i.e. formulas that you would like to program in FPScript, you should carefully plan this structure. The following elements are available to help you build your analysis network:

**Analysis Object:** If an analysis object is available for your analysis, it is recommended that you use it. Analysis objects have an advantage over conventional FPScript formulas in that you can set parameters via a dialog box, and the physical unit of the result can often be determined automatically.

**Formula:** Calculates a result that you would like to display or that you need for further calculations.

**FPScript Function:** Calculates a result for one or more arguments returned when calling the function.

To structure the analysis, use the following procedures:

1. First plan an analysis object or a formula per result that you would like to display.
2. Analyze each formula that you planned for multiple instances of code elements, which are completely identical and which calculate a single result in one of the FPScript data structures. For these types of code elements, create additional formulas whose results

you then access in the formulas already present.

3. Now you can analyze your code for recurring elements that use different data, but otherwise have the same structure and calculate a single result in one of the FPScript data structures. For these types of elements, create FPScript functions that you call from existing formulas and that return the data to be included as arguments.

A carefully planned analysis should be as efficient as possible and should not have redundant code elements. Avoid redundant code elements by "outsourcing" them in separate formulas or FPScript functions. You can also achieve efficiency by avoiding unnecessary loops in the FPScript code.

### Using the list data structure correctly

As described above, each formula should return as its result exactly one of the final results to be displayed. In rare cases, however, it is impossible to separate calculation of several results without considerable loss of efficiency. If an algorithm returns, for instance, several results, then in principle it is always possible to separate the calculation of the individual results by using the same algorithm in one formula for each result. However, this leads to repeated processing of the algorithm. In such cases, you should do the following:

1. Create a formula using the appropriate algorithm to calculate the results.
2. Return all results as a List data structure. Name the elements of the list according to the calculated results.
3. If the results in the list have a different data structure or physical unit, for each result, create an additional formula that extracts a list element from the list. On the **General** tab of these formulas, enter the respective units and, if applicable, comments. You should name the formula the same as the list element that you extract.

### Example

**WaveAnalysis** : A formula that breaks down a given signal into individual waves and, for each individual wave, the RMS value and the harmonic distortion (THD=Total Harmonic Distortion). The result is a list with a data series with the RMS values and a data series with the THD values.

**THD** : A formula with the unit % that returns the THD value.

**RMS** : A formula with the unit V that returns the RMS value.

## Analyzing Multiple Measurements

Analyzing multiple, similar measurements is one of the most important automation tasks in measurement applications. Since FlexPro manages each analysis as a dynamic object network, you only need to "exchange" the data sets upon which the analysis is based.

To do this, you set up several folders in the project database. Each folder contains the respective data sets from a certain measurement. FlexPro automatically creates these folders when importing measurement data files if in the **Import** dialog box you enable the option **Create new folder for each file**. Here, it is important that the data sets from the various measurements are always given the same name. You create the documents and analysis objects in the parent folder.

Now for example, before you update a document and thus trigger a calculation, activate the subfolder from which the data is to be taken by selecting it in the **Activate or deactivate folder** drop down list at the top of the Folders window. All of the objects in the activated folder are then displayed in the parent folder. FlexPro first searches in the activated folder for all data sets referenced in formulas, curves, etc., and then, if they are not found there, it searches within the parent folder.

For this to work for all references to the data, only the name and not the entire path name of the data set must be entered. In the curve, for example, **Signal** should be entered as the data set and not **\Measurement1\Signal**. In the latter case, this would mean that the data set would continue to be taken from **Measurement1** after the folder **Measurement2** is activated. You can make sure that this is the case by activating a measurement folder already before creating the analysis and documentation and then selecting the data sets blended-in, e.g. when creating a diagram. FlexPro recognizes that a simple name is enough to refer to the data set and then only enters the name as a reference.

In the FlexPro installation folder you can find the **Examples** folder, and in this folder is the **Examples** project database. In the project database there is an **Analysis** folder where you can find an example covering the topic above.

The project database path name under Windows Vista or Windows 7 is usually C:\Users\Public\Documents\Weisang\FlexPro\9.0\Examples\Examples.fpd or C:>Users>Public>Public Documents>Weisang>FlexPro>9.0>Examples>Examples.FPD, and under Windows 2000 or XP it is C:\Documents and Settings\All Users\Shared Documents\Weisang\FlexPro\9.0\Examples\Examples.fpd. If you cannot find the project database on your computer, you may have to install the database. To do this, start the Setup program on the FlexPro CD, select **Modify** and on the next page select the category **Examples** and then **Example Databases**.

## Efficient Data Analysis

The FlexPro's data analysis is based on the powerful FPScript analysis programming language. In addition to the optimum setting of the FPScript runtime environment and the FlexPro object database, using FPScript correctly plays a decisive role in efficient analysis. Avoiding loops is the most important factor when creating an analysis algorithm using FPScript. To achieve this, the following FPScript features are available:

- FPScript offers the ability to calculate complete data sets in simple arithmetic expressions.
- For all common analyses, FPScript offers a variety of integrated functions.
- The event isolation functions localize events such as extreme values, slopes and level crossings.
- The Index operator extracts individual values or ranges of data.

With the above information to help you, you can almost always avoid loops, especially across all values of a data set. If a loop over individual values cannot be avoided in FPScript, then you should use the For Each Value...End statement, which is the fastest of all FPScript loops.

### Example

For individual waves from a power signal, the RMS value should be calculated and output as a data series. This task can be solved as follows:

1. With the LevelCrossings function, the positions (indices) of all positive zero crossings in the signal are determined as data series.
2. Using a For loop, run across all values in the data series.
3. For every two neighboring positions with the Index operator, take the range between these two indices from the signal.
4. For each signal segment formed this way, calculate the square mean (RMS).
5. With the Append operator, collect the RMS values calculated this way in a data series.

The example above uses a loop, not across the individual values of the data set, but instead only for the periods that occur therein. The loop is necessary because even if the signal has a constant sampling rate and frequency, the length of the individual data ranges can vary slightly.

1. In the special case that the signal was sampled with phase synchronization, e.g., with 1000 values per period, the calculation can be simplified considerably:
2. With the NextLevelCrossing function, search for the first positive zero crossing and take the part of the signal starting at this position using the index operator to remove

the incomplete wave at the beginning.

3. With the Reshape function, transform the data series with the N amplitude values into a data matrix with  $N / 1000$  columns and per 1000 rows. Any incomplete wave that might occur at the end of the signal will then be rejected.
4. For this data matrix, calculate the RMS values column by column using the Mean function.

You can find both examples in the FPScript Example database in the **Wave Analysis** folder.

## 5.2 Working with Analysis Templates

### Creating an Analysis Template

1. Select all of the objects in the Object List that you want to save as an analysis template.

---

**Note:** It is sufficient if you select only the objects at the highest hierarchical level. FlexPro automatically adds linked objects to the template.

---

2. Now right-click with your mouse on a selected object and choose **Save as Template**.
3. On the first screen of the **Template Wizard** that appears, choose **Analysis** as the **Template type**.
4. Follow the wizard's next steps and use Help in the wizard if you require additional assistance.

### See Also

Analysis Templates

Organizing Analysis Templates

Template Databases

### Using an Analysis Template

1. Select all of the objects from the Object List that you want to use for an analysis.
2. Select **Analysis** from the FlexPro **New** menu.
3. Follow the steps of the **Analysis Wizard** and use the wizard's Help if you require additional assistance.

**See Also**

Analysis Templates

Creating an Analysis Template

Organizing Analysis Templates

Template Databases

**Organizing Analysis Templates**

Use the **Organizer** dialog box to delete analysis templates or to copy templates from one data base to another.

**Opening the Organizer dialog box and selecting the the template type**

1. In the FlexPro **Tools** menu select **Organizer**.
2. In the **Organizer** dialog box click on the **Templates** tab.
3. In the **Template type** list box select **Analysis**.

**Opening the current project database or a template database for organization**

- Choose a template database from one of the **Items available in** list boxes.

**Opening a shared template database or any other database for organization**

1. Click on one of the **Close Project Database** buttons to close the database currently displayed in the list.
2. Click on **Open Project Database** and select any project database on your hard disk or on the network.

**Copying templates from one project database to another**

1. Select the templates that you would like to copy.
2. Click **Copy**.

**Deleting templates from a project database**

1. Select the template that you would like to delete.
2. Click **Delete**.

## 5.3 Formulas

A formula is a special data set where the value is not fixed when stored, but is instead replaced by calculation operations that can calculate the value when required. These formulas are written in FPScript. FlexPro offers you special analysis objects for common analyses so that you do not have to learn how to use FPScript.

Formulas are the main tools for automating the analysis. In the nested network of formulas for a project, the entire "knowledge" is saved so that you can repeat the analysis as often as you like, and the data to be analyzed can be assigned at the click of a button.

You can use formulas wherever data sets are also permitted, such as for a curve in a diagram.

FlexPro compiles the formula into binary code, which guarantees rapid processing. The FlexPro "Smart Update" technology ensures that unnecessary re-calculations are avoided. The result of the last calculation is stored temporarily and re-used for as long as possible.

---

**Note:** With FlexPro View, the analysis options are limited to basic mathematical operations and simple, statistical evaluations.

---

### Formula Elements

- **Constants:** These are constant expressions such as values in the various data types.
- **Variables:** You can use variables in a formula to save interim results. This allows you to save computing time by only performing a calculation once for terms that are used several times. You can also structure your formulas better by first assigning parameters to a variable and then continue on with the formula using the variable.
- **Data sets:** Data sets are used like variables. If a data set name is used in a formula, this name represents the data that is read when the formula is later calculated. This data can also be formulas. The formulas are then first evaluated to obtain the data. You cannot make assignments to formulas and data sets.
- **Pre-defined constants:** Some mathematical constants, such as PI or E, as well as constants, which are used as function parameters, are available as pre-defined constants. You cannot use these names for data set names and/or variables.
- **Operators:** You can use operators to calculate and modify data. FlexPro provides you with a variety of operators with which you can also calculate complete signals.
- **Statements:** Each formula consists of one or more statements, where one line is used for each statement. However, you can also write several statements in a single line, if



you separate the statements from each other using a semi-colon ';'. You can also split a statement over several lines. For lines that are to be merged with the lines following, you have to enter a backslash '\' or an underscore character "\_" as the final character. In FlexPro formulas you can use different loops and conditional statements, for example, to be able to perform a complex analysis for all of the data series in a signal series.

- **Functions:** FlexPro features a variety of functions that you can use to generate signals and carry out analyses. You can also define custom functions. These are formulas that you provide with arguments and that you can call like a function in other formulas.
- **Object properties:** The various objects in FlexPro provide special functions for accessing their attributes. For example, you can access the comments for a data set.
- **Comments:** These start with "//" and then extend to the end of the line.

### Examples of Formulas

The project database **Examples** is located on your hard drive and contains various FPScript programming examples.

The path to the project database is usually C:

>Users>Public>Documents>Weisang>FlexPro>10>Examples>Examples.fpd or C:

>Users>Public>Public Documents>Weisang>FlexPro>10>Examples>Examples.FPD.

If you cannot find the project database on your computer, you may have to install the database. To do this, use the **Custom** setup type and select **Example databases**.

## Working with the Formula Debugger

### Creating a Formula

1. In the Folders window, select the folder in which you want to place the new formula.
2. If necessary, select from the Object List data sets, other formulas or a data range in the data view or a data editor window. A corresponding statement for accessing the data selected is then inserted into the new formula.
3. Select **Insert Formula** from the **FlexPro Objects** toolbar. A new formula is created that you can edit.

## Editing a Formula

Before you can edit a formula, you first have to open it. To do this, double-click on the formula in the Object List. A window appears where you can then edit the formula. For the formula code, use the FScript scripting language.

The window for editing formulas has two views. The formula is edited in **FScript** view, and in **Value** view, the current result of the formula is displayed numerically. The functionality of the **Value** view corresponds to that of a read-only data set window.

The formula editor works like a common input box. Use the following keys to navigate:

Key	Function
CURSOR KEYS	Move the cursor in the corresponding direction.
CTRL+RIGHT	Moves the cursor to the beginning of the next word.
CTRL+LEFT	Moves the cursor to the beginning of the previous word.
HOME	Moves the cursor to the start of the line.
END	Moves the cursor to the end of the line.
CTRL+HOME	Moves the cursor into the first line.
CTRL+END	Moves the cursor into the last line.
TAB	Indents the selected lines.
SHIFT+TAB	Unindents the selected lines.

---

**Note:** If you hold down the SHIFT key while navigating, the passage you move over is selected.

---

## Inserting Code Elements into a Formula

For all common elements that you use to construct your formula, FlexPro offers wizards that free you from a large part of the coding workload. There are wizards for inserting functions, object properties, operators, constants, statements, data sets and index operations.

### Inserting a Function

1. Place the cursor at the position where you would like to insert a function.
2. Press CTRL+SPACE to view a list of available functions.
3. Select the desired function from the list and press ENTER to insert it into the formula.

4. The FlexPro Assistance Window will then appear, which will assist you in entering the function arguments. Another list appears for data arguments that contains data sets in the folder of the formula whose data structure is valid for the argument. For arguments requiring a constant, a corresponding list appears with the valid constants. Use the UP key to switch to the list and to select an entry.
5. After selecting an argument, add a comma, if applicable, to enter additional arguments.

---

**Note:** To view the Assistant Window for the argument at a later time, place the cursor on the argument and press SHIFT+CTRL+SPACE.

---

### **Inserting an Object Property**

#### **a) Using the Formula Editor's Assistance function**

1. Next, specify the object by entering one of the keywords This, ThisFpObject, ThisObject or Application or an object reference, such as '\2D-Diagram.2D'.
2. Now enter a period to view the list of available properties.
3. Use the DOWN key to switch to the list and to select an item from the list. Transfer the item using the ENTER key.

#### **b) Using the Object Properties Wizard**

1. Right-click with your mouse on the position where you would like to insert an object property.
2. In the context menu, choose **Insert object property**.
3. A wizard appears to assist you with the remaining steps.

### **Inserting a Unit Symbol**

1. Right-click with your mouse on the selection or position where you would like to insert a unit symbol.
2. In the dialog box, select either the required unit symbol directly or first select the physical quantity and then the appropriate unit symbol.
3. Click OK to confirm and apply your selection.

### **Inserting Data Sets from the Object List**

Data sets, formulas and analysis objects that you want to use for the formula calculations can be transferred from the Object List into the formula. Simply drag the desired data set with the mouse from the Object List to the position in the formula

where the reference is to be inserted. A reference to the object is then placed in the formula. If the inserted object is located in the same folder or in an activated subfolder, only the name is entered. Otherwise, the path name of the object is entered. The object has to be located in the same project database as the formula. You can also drag a cell range selected in the data view of a folder or a data set window into the formula. An FPScript expression is then inserted that extracts the selected section from the data set.

### Search and Replace in a Formula

You can search for text passages in a formula and replace these passages with other text. Select the command **Find** or **Replace** from the **Edit** menu. Dialog boxes then appear where you can specify the required text. Use the **Find Next** command to run the last search in each case several times.

### Debugging Formulas

After you have written a formula, you should check that the formula works correctly. Three types of errors can occur in formulas: syntax errors, runtime errors and semantic errors.

#### Types of Errors

A syntax error occurs when you violate the syntax of the FPScript language, such as if you have written a keyword incorrectly. FlexPro recognizes this type of error as soon as the formula is compiled into binary code. The **Syntax Check** command in the Formula menu lets you check the correctness of the syntax in a formula at any time. If there is a syntax error, this is displayed in the Event Log, and you can select the defective section of code by double-clicking on the message.

A runtime error occurs when there is a problem during calculation of the formula, such as if you were to divide by zero. The calculation of the formula is then canceled and the error is displayed in the Event Log. You can select the defective passage in the code by double-clicking on the message. Use the **Run** command in the **Formula** menu to calculate a formula on a test basis and thus check whether runtime errors occur. The result of the calculation is displayed in a window.

A semantic error occurs when a formula does not work as the programmer expected. These types of errors are often difficult to find because FlexPro cannot localize them on its own. However, FlexPro offers you the Formula Debugger that you can use to step through the formula and analyze all of the interim results.

## Formula Debugger

The Formula Debugger is a powerful tool that you can use to step through and test formulas. Once started, the debugger can be in Single Step mode or in Run mode. You can execute single statements in single step mode and review their results as well as the contents of local variables. In run mode, the formula is processed at normal speed until a breakpoint is reached, it is aborted by you, or the end of the formula is reached.

FlexPro displays the content of selected variables and the result of the statement just executed in the Watch Window while you are debugging a formula.

## Working with Formulas

### Starting the Debugger

**To perform controlled execution of a formula until you reach a breakpoint using the debugger:**

1. Open the formula which you want to execute under control of the debugger.
2. Click on **Debug/Continue** in the **Formula** toolbar.

---

**Note:** The formula is executed immediately and ends when the end of the formula or a breakpoint is reached. You can cancel its execution at any time by pressing **Break** in the **Formula** toolbar.

---

**To step into a formula using the debugger:**

1. Open the formula which you want to execute under control of the debugger.
2. Click on **Step Into** in the **Formula** toolbar.

---

**Note:** The debugger marks the first statement in the formula, but will not start executing it.

---

### Stepping into Statements

The debugger must be in Single Step mode. If the bugger is in Run mode, click on **Break** in the **Formula** toolbar.

**To execute the highlighted statement and an FPScript function called within it:**

- Click on **Step Into** on the **Formula** toolbar.

**To execute the entire statement hilighted:**

- Click on **Step Over** on the **Formula** toolbar.

**To continue execution until the next breakpoint:**

- Click on **Debug/Continue** on the **Formula** toolbar.

### Setting and Removing Breakpoints

**To set a breakpoint:**

1. Open a formula in which you want to set a breakpoint.
2. Set the cursor on the line in the formula that contains the statement before whose execution the debugger is to be switched to Single Step mode.
3. Click on **Toggle Breakpoint** in the **Formula** toolbar. A breakpoint symbol appears at the left border of the line.

**To remove a breakpoint:**

1. Open the formula containing a breakpoint that you want to remove.
2. Set the cursor on the line in the formula that contains the breakpoint.
3. Click on **Toggle Breakpoint** in the **Formula** toolbar.

**To remove all breakpoints from a formula:**

1. Open the formula containing all breakpoints that you want to remove.
2. Click on **Remove All Breakpoints** in the **Formula** toolbar.

**Monitoring Variable Content**

The debugger must be in Single Step mode. If the debugger is in Run mode, click on **Break**.

**To display the content of a local variable in a popup window:**

- Click on the name of the variable in the formula.

**To display the content of a local variable in the Watch Window:**

1. Right-click on the name of the variable in the formula.
2. In the context menu, select **Add Watch**.

**To select the variable to be displayed in the Watch Window:**

- Select it from the Watch Window drop down box.

**To remove a variable from the Watch Window:**

1. Select it from the Watch Window drop down box.
2. Click on **Delete Variable** next to the drop down box.

**FPScript**

FPScript is a programming language that is particularly easy to learn. You can use this language to program your own analyses. Use FPScript when you want to program your own formulas or functions or as embedded FPScript for accessing data or attributes in presentation objects.

The FlexPro analysis objects are also based on FPScript. However, if you use an analysis object, you do not have to program the FPScript code yourself. Simply configure the analysis via the analysis object's Properties dialog box and FlexPro automatically creates the appropriate FPScript code.

FPScript has been developed specifically for technical/scientific data analysis and offers powerful operators that you can use to calculate individual values as well as entire data series, signals, etc. in a single line. FPScript formulas can provide each of the data structures and data types valid for data objects as the result. In particular, aggregate data structures consisting of up to three components can be calculated in a single expression.

FPScript not only uses numbers for calculations, but also physical quantities composed of the value and unit. During arithmetical operations the units are adjusted automatically, if

necessary, and the unit of the result is derived correctly from the units of the arguments. This eliminates the need for using scaling factors in FPScript expressions as they are required to adjust units when using purely numeric programming languages. Another source of errors is in calculating incompatible units. FPScript recognizes this and outputs an error message.

FlexPro handles formulas that you have written using the FPScript language just like data sets. For instance, you can display the results in diagrams or tables or calculate them in additional formulas or analyses. Therefore, the most common statements in FPScript programs are arithmetical expressions that you can use for calculations. As opposed to common programming languages such as Basic, FPScript allows you to calculate whole data series, signals etc., in addition to scalar values, in one single statement. You do not have to use loop constructs to do this.

### Arithmetical Expressions

An arithmetical expression consists of operators, which determine the operations, and operands, which supply the data to be calculated. Most operations have two operands. The operator is then written between the operands. For instance, to add two figures, write:

```
1.5 + 2.3
```

For the following expression, you would have to program a loop already if you were using Basic:

```
DataSetSeries * 2
```

**DataSetSeries** is the name of a data set that could, for instance, contain a series of measurements with 10,000 measurement values. This is multiplied by a factor of 2. FPScript carries out this multiplication on a per-element basis, which means that each value in **DataSetSeries** is multiplied by 2. The result of the expression is then once again a data series with 10,000 values.

Expressions with the following structure are also very common:

```
DataSetSeries1 + DataSetSeries2
```

Here, two data series are added. Again, FPScript proceeds on a per-element basis, i.e. the first value from **DataSetSeries1** is added to the first value from **DataSetSeries2**, etc. The result is once again a data series. The two operands should have the same number of values. If this is not the case, the number of calculations matches the number of values in the shorter data set.

You can also string several operations together in a single expression; for instance:

```
2.5 + 3 * DataSetSeries
```

The above example shows linear scaling of a data series. The Y axis division is 2.5 and



the gradient is 3. If operations are strung together, operator priority and associativity determine the order in which the individual operations are carried out. Multiplication always takes precedence over addition. Therefore, FPScript first multiplies the `DataSet` by the value 3 and then adds 2.5 to the result. If you would like to determine the order of calculation yourself, you have to use parentheses, as you would in standard mathematics:

```
(2.5 + 3) * DataSet
```

In this case, all values in `DataSet` are multiplied by the value  $(2.5 + 3)$ , which equals 5.5.

The associativity of operations plays a role if several operations with the same priority are strung together. Most operations are left-associative, meaning that the operations are carried out from left to right:

```
DataSet1 + DataSet2 - DataSet2
```

corresponds to the following expression using parentheses

```
(DataSet1 + DataSet2) - DataSet2
```

Some operations only need one operand. In this case the operator is written before the operand:

```
-DataSet
```

negates all values in `DataSet`, for instance.

## Quantities in FPScript

### Physical Quantities

One of FPScript's strengths is the ability to perform calculations using physical quantities composed of a value and unit. FPScript not only manages the unit symbol, but also the SI dimension, which makes it possible to check and, if necessary, transform different units for compatibility.

### Entering Quantities

By appending a unit to a constant, the constant becomes a **Quantity**. The value of a quantity can be of the 32-bit or 64-bit floating point data type and real or complex. Quantities of the integer data type are not supported.

The unit is written after the constant and separated by spaces, such as in:

```
1.3 N
```

Degrees ( $^{\circ}$ ), minutes ( $'$ ) and seconds ( $''$ ) are an exception. These units are written

immediately after the numeric value without spaces:

15° + 30' + 45''

or

15° + 30' + 45"

New units can be formed from most units using an SI prefix:

1 kHz

1 MHz

1  $\mu\text{m}$

FPScript also accepts the notation frequently used with u instead of  $\mu$  for the prefix "micro":

1 um equals 1  $\mu\text{m}$

However, u is written alone as the symbol for interpreting the atomic mass unit:

1 u equals 1 Da equals 1 Dalton

FPScript accepts the symbol or the name of a unit. Symbols are case sensitive, but case is irrelevant for names. The following are correct forms of notation:

5 V

5 mV

5 Volt

5 volt

5 Millivolt

2  $\Omega$

2 Ohm

The following, on the other hand, are not accepted:

5 mVolt

5 Volts

Derived units can be constructed as a product of powers of known units.

1 N m

1 N\*m

1 N·m

Exponents are introduced with the ^ character:

1 m s<sup>-2</sup>

For the exponents 2 and 3, FPScript also accepts the superscript numbers:

1 m<sup>2</sup>

1 m<sup>3</sup>

Instead of negative exponents, you can also use the / character (division):

1 m/s<sup>2</sup>

1 1/min

Make sure to use parentheses correctly:

1 m/(s kg)

If you use **Unit monitoring** on the **Unit Manager** tab of the project database **Properties** dialog box and set it to **Tolerant**, you can omit the empty spaces or multiplication sign between the basic units:

1 Ws

However, the result of this is that it could be interpreted incorrectly, such as in the following:

5 Volts

is then accepted as

5 V·o·l·t·s

equals 5 volts·octet·liter·ton·second.

---

**Note:** You can avoid misinterpretations by using unit symbols only and separating them with spaces. Use  $\mu$  and not u for the "micro" prefix. Check your use of upper/lower case in the prefix and unit symbol. Typical errors include S (Siemens=conductance) instead of s (seconds=time) or M (mega= $10^6$ ) instead of m (milli= $10^{-6}$ ).

---

You can also use rational exponents. The following example depicts the franklin from the Gaussian unit system in SI units:

1 g<sup>(1/2)</sup> cm<sup>(3/2)</sup> s<sup>-1</sup>

Units may contain a prefactor and an offset. The following depicts the unit ° Fahrenheit in the Kelvin SI unit:

1 0.5555555555555555K+255.372222222222

In the event of ambiguities, you will have to place the unit between apostrophes:

Dim s = 2

v = 3 'Vs'\*s

This is also required for a negative prefactor:

5 '-1.602176487E-19 C'

## Checking the Unit Symbol

FlexPro breaks a unit down into its elements during compiling and stores the unit symbols of the elements individually with their corresponding exponents. In addition, the unit symbol as specified is saved as a custom unit symbol. For

```
1 N m/(1/min)
```

the custom symbol "Nm/(1/min)" is saved with the exponents 1, 1, 1 as the symbols of the basic units "N", "m", "min". When calculating different units, the custom symbol is discarded and the result unit symbol is reconstructed from the symbols of the basic units:

```
1 N m/(1/min) * 1 min^-1
```

results in:

```
1 N m
```

You can replace the custom symbol with the `ChangeUnitSymbol` function without affecting the unit. You can also use the same function to delete it. FlexPro will then regenerate the unit symbol from the basic units. For the example above, the result is "N m min".

## Using Quantities for Calculations

FPScript adjusts the units to each other before processing the quantities. The following FPScript expression

```
1 V + 2 mV
```

provides the result 1.002 V, for instance, and not the value of 3, which would provide a calculation that does not account for the unit. The right operand is transformed to the unit of the left operand prior to addition.

The same applies to comparison operations:

```
5 mV < 1 V
```

provides TRUE, for instance, and not FALSE.

Another frequent source of error when performing purely numeric calculations of quantity comparisons is calculating incompatible units. The expression

```
1 A + 2 V
```

generates an error message in FPScript, since the units Ampere and Volt have different SI dimensions.

```
1 km + 5 NM
```

is permitted, however, since the nautical mile can be converted to km.

If the values, i.e. pure numbers are calculated with quantities, the value then implicitly takes the unit of the quantity:

```
1 V + 2 = 2 + 1 V = 3 V
```

This basic principle, which applies to comparison and other operations as well as to function arguments, makes it easier to write "unit-neutral" FPScript code.

For some operations, the unit of the result is formed from the units of the arguments:

```
1 m / 2 s
```

provides 0.5 m/s as the result.

```
(1 m) ^ 2
```

provides 1 m<sup>2</sup> as the result.

For vector data, i.e. data series and data matrices, all elements always have the same unit. The individual components of an aggregate data structure can, however, have different units, thus, for instance, the Y component of a signal can have the unit Nm and the X component can have the unit s.

The units are accounted for not only in FPScript operators, but also in all FPScript functions.

```
Derivative(Path)
```

provides, for instance, the speed in the unit m/s for the case in which **Path** is a time signal with the X unit s and the Y unit m.

Depending on the unit monitoring setting, FPScript also permits unknown units for the result, although they cannot be transformed. For instance:

```
1 Dummy + 3 Dummy
```

results in 4 Dummy.

The value of a quantity determines the Value operator:

```
Value 5 mm
```

returns the 64-bit floating point value 5.

With the Unit operator you can transform the unit of a quantity:

```
Unit<mm> 2 cm
```

provides 10 mm as the result.

---

**Note:** FlexPro's behavior with regard to processing units depends on the **Unit Manager** tab of the project database **Properties** dialog box **Unit monitoring** settings.

---

## Unit Calculation

With the Unit function you can extract the unit of a quantity:

```
Unit(5.3 V)
```

```
Unit({56 V, 7.8 V})    // A data series with two values is returned here
```

Making these two calls provides 1 V as the result.

Thus, you can carry out **unit calculations** and use the AdjustUnit function to transform a quantity to the result unit:

```
AdjustUnit(Frequency, 1 / Unit(Time))
```

The unit of **Frequency** is transformed to the reciprocal value of the unit of **Time**.

## Percentage Calculation with the Units % and ppm

The basic types of calculation, which are addition, subtraction, multiplication and division, are calculated for operands with the unit % or ppm on the right side in a special way:

$$200 \text{ m} + 10 \% = 200 \text{ m} + 200 \text{ m} * (10/100) = 220 \text{ m}$$
$$200 \text{ m} - 10 \% = 200 \text{ m} - 200 \text{ m} * (10/100) = 180 \text{ m}$$
$$200 \text{ m} * 20 \% = 200 \text{ m} * (20/100) = 40 \text{ m}$$
$$200 \text{ m} / 20 \% = 200 \text{ m} / (20/100) = 1000 \text{ m}$$

For the unit ppm, the calculation is carried out in a similar manner with the factor 1.000.000 instead of 100.

---

**Note:** The percentage calculation must be enabled on the **Unit Manager** tab of the project database **Properties** dialog box. Otherwise, the units % and ppm are treated like all other units.

---

## Exception Handling

FPScript exception handling allows you to react flexibly to errors in your formulas. Exceptions are all errors that may occur during the execution of FPScript code as well as exceptions that have been explicitly generated using the Throw statement.

The following example carries out a division calculation and generates an exception if the divisor equals zero.

```
Arguments X, Y
If Y == 0 Then
    Throw "Divide by zero"
End
X / Y
```

With the Try...Catch statement, you can catch these types of exceptions and react accordingly. In the case of an error in a calculation, for instance, you can use an error message to prevent the calculation from being aborted. Instead, you can catch the error and, for example, return an empty result:

```
Try
    Return Integral(x)
Catch Exception
    Return Empty
End
```

The variable specified behind the Catch statement is assigned the "thrown" value when generating the exception using Throw. This can be of any data structure. The functions and operators built into FPScript pass a string with a description of the error as a value of the exception.

Exceptions are a powerful concept for error processing by distinguishing between a regular result and an exception. You should, however, only use exceptions to handle errors and not as an alternative for passing results. The reverse is also true. You should not use the formula's resulting value to pass error messages, but generate an exception for this instead.

## Functions

FPScript offers you a host of built-in functions. You use a function by writing the name, followed by a list of arguments, which is to be specified in parentheses, into the formula. The following example integrates a data set:

```
Integral(DataSet)
```

If you have to specify several arguments, these are comma-separated:

```
Expand(DataSet, 2)
```

Even for functions that do not need an argument, the argument list parentheses must be included:

```
CurrentDate()
```

Some functions have optional arguments. In most cases, the argument list is created in such a way that the optional arguments are at the end of the argument list and can be omitted:

Mean(Signal), for instance, is the equivalent of Mean(Signal, MEAN\_ARITHMETIC +

PROCESS\_ROWS).

For some functions, however, an optional argument can be omitted in the middle of the argument list:

Noise(1, , 0), for instance, is the equivalent of Noise(1, NOISE\_NORMAL, 0).

Functions account for the units of their arguments and also provide quantities as the result:

LevelCrossings(Height, 25 cm) also returns the correct result, for instance, if **Height** is in the unit m and not, like the threshold value, in the unit cm.

---

**Note:** Only a small selection of functions is available in FlexPro View.

---

### Results of Formulas

When a formula is processed, the result of the last statement is saved as the result of the formula and returned to the caller.

The following example generates a signal with a sine wave in several steps:

```
Dim x, y
x = (100, 0 s, 0.1 s)           // Create data series with 100 increasing time values
y = 5 V * Sin(2 Pi * 3.5 Hz * x) // Sine with frequency of 3.5 Hz and amplitude of 5 V
Signal(y, x)                    // Combine X and Y to form signal
```

The formula result is the value that is returned by the Signal function.

If you do not want to pass on the result of the last statement, you must use the Return Statement:

```
If NumberOfRows(Signal) < 2n Then
    return ?           // If Signal has less than 2 values, then pass void floating point
End
1. / (Signal.x[1n] - Signal.x[0n]) // Otherwise, calculate sampling rate
```

FlexPro very rarely re-calculates formulas. It ensures that each formula is only calculated once during an update.

If there is already a result present from the previous update procedure, FlexPro first checks this to see if it is necessary to do a recalculation. This is the case when the formula is based on data that has changed since the last calculation or the formula uses functions with a result that is deterministic, e.g., the random function. If the existing result is still valid, then it is re-used without a recalculation.



## Extracting Data

In addition to calculating data and creating data structures, extracting parts of data sets is an important use of FPScript; for instance, all local maxima in a signal or the range between two zero crossings. For these types of applications, FPScript offers two very powerful index operators.

## Index Operator

The Index Operator extracts individual values or ranges of values using their position in the data structure. All rows in data series and all rows and columns in data matrices are numbered starting with zero. Negative indices are counted from the end, which means that -1 indexes the last row or column. The index operation makes it possible for you to access individual values or ranges using these numbers.

`DataSet[99n]` reads, for instance, the hundredth value from a data series.

`DataSet[-2n]` reads the second to the last value from the data series.

`DataSet[0n, 49n]` creates a data series with the first 50 values from `DataSet`.

`DataSet[49n, -1n]` creates a range from the 50th value to the last value of a data series.

`DataSet[{0n, 5n, 7n, 9n}]` creates a data series with the values that have indices listed. Here, the Bundle operator was used to bring together the required indices into a data series.

`DataSet[(NumberOfRows(DataSeries), NumberOfRows(DataSeries) - 1n, -1n)]`

Here, the data series operator is used to generate the indices. The statement reverses the order of values in the data series.

For a data matrix, specify two indices. The first index selects one or more columns and the second selects one or more rows:

`DataMatrix[1n][2n]` reads, for instance, the third value from the second column of the data matrix as a scalar value.

`DataMatrix[-1n]` reads the last column from a data matrix as data series.

`DataMatrix[][2n]` reads the third row from a data matrix as a data series.

## Value Index Operator

When working with signals and signal series, it often makes more sense to specify the X and Z values directly when indexing sections rather to work with the indices. To do this, FPScript provides the Value Index Operator.

`Signal[[0 s, 4.5 s]]` creates, for example, a partial signal with the points of the first

4.5 seconds.

Note that the operation functions as expected even when the X component of the signal has the unit ms, for instance.

`SignalSeries[[3.5 kHz]]` extracts the signal with the Z value 3.5 kHz from the signal series.

The syntax of the Value Index operator corresponds to that of the Index operator. All you have to do is to include all square brackets twice. FPScript then interprets the indices specified as X or Z values and looks for these in the X or Z component of the signal or signal series in order to obtain the positions.

### List Element Operator

You can use the List Element Operator to extract one or more elements from a list, allowing you to work with the list element indices or names.

`List.[1n]` provides, for instance, the second element in the list.

`[-2n]` takes the second to last element from a list.

`List.Name` takes the primary element called "Name" from a list.

`List.[1, 2]` returns a partial list with the 2nd and 3rd element.

`List.["Strom*"]` returns all list elements whose names start with "Strom".

### Accessing Object Properties

The objects saved in the project database, such as documents or diagrams, have a series of properties. For example, each object has a name or a creation date; data sets have a physical unit, etc. You can use FPScript for read access to these types of object properties. You can, for instance, query the unit name of a data set or the current cursor position in a diagram. To do this, use the following syntax:

`[Object].Property[(Parameter1,..., Parameter n)]`

Object is the path name of the object and Property is the name of the property that you want to access. If the object is not a data object, i.e. a data set or a formula, then you have to append the file name extension describing the object type to the object name. If you do not specify an object name, then the formula containing the FPScript code is accessed. To access certain objects, which depend on the context of the processed FPScript formula, use the `This`, `ThisFPObj` or `ThisObject` keywords.

`Diagram.2D.Name` supplies the name of the 2D diagram **Diagram** as a string. **Name** in this case is the name of the property to be read and **.2D** is the file extension for the 2D diagram.

`.\FullName` provides the absolute path name for the folder where the FPScript formula is located.

`\Comments` provides the comments of the project database root folder.

`.Unity` provides the physical unit of the Y component of the formula containing this FPScript expression.

You can also access object properties using a variable if you have already assigned an object reference to it:

```
Dim Obj = MyDataset As Object  
Obj.CommentsX
```

Write-access to object properties is greatly reduced and only permitted for the header information of the formula containing the FPScript code. It is, however, preferred that you handle the corresponding entries in the Properties dialog box of the formula for these properties instead of writing these in the FPScript code. During each update process, formulas that use this type of write access are not considered up-to-date and thus must be recalculated each time. It makes more sense to have write-access to object properties in the embedded FPScript.

Instead of directly specifying the object path name or the property name, you can use the indirection operator, which uses a string containing the name. The parameter list is only required for properties that require additional parameter settings. To access the cursor position in a diagram within a worksheet, for example, you have to specify the number of the plane and the number of the cursor.

FPScript allows read access to all object properties that are defined in the FlexPro automation object model. It is used very frequently in embedded FPScript formulas, which are used, for instance, to label the axes in a diagram. Within FPScript formulas that you use for data analysis, such access is mainly targeting attributes and parameters of objects as well as cursors and markers in diagrams. You can access the FPScript expressions required for this through the Object Properties Wizard of the Formula Editor.

## **Accessing Data and Objects**

Not only can you access local variables in FPScript, but you can also access external data sets and calculation results. If you use a name such as "DataSeries", FPScript proceeds as follows to find the object with this name. Initially, FPScript checks to see whether a pre-defined constant exists with this name. It then checks to see whether a local variable has been defined with this name in the formula. If neither is the case, and if the FPScript formula is a function with arguments, it checks to see if an argument variable exists with this name. If nothing is found during the search, it searches for a data set or a formula with this particular name. The search takes place in the folder containing the formula. If, however, this folder has an activated subfolder, this subfolder takes priority in the

search. If the object found is a formula, this is calculated, if necessary, and the result is taken for further calculation.

### Path Name

The object path name describes its precise location within the project database.

**Absolute path names** provide the path from the root folder of the project database to the required object. Folder names are listed as a series, where each folder name ends with a '\'. Since the root folder does not have a name, only a '\' is written for it. Therefore, all absolute path names start with a '\'. The following example locates the object called "DataSet" in the "Measurement1" subfolder of the "Analysis" folder:

```
\Analysis\Measurement1\DataSet
```

The "Measurement1" subfolder is selected this way:

```
\Analysis\Measurement1
```

This access provides all data sets within the folder as a list in the order in which they are displayed in the unsorted object list.

---

**Note:** If in the example above a data set called "Measurement1" is in the "Analysis" folder, access will be provided to this data set and not to the folder of the same name. To access the folder in this case as well, you need to add the '.fld' extension.

---

An object called "DataSet" located in the root folder is selected as follows:

```
\DataSet
```

If there are special characters, e.g., spaces, in the path name, or if it begins with a number, then you must specify the path using single quotation marks:

```
'\Analysis\Measurement A\DataSet'
```

```
'\01\DataSet'
```

If objects are to be accessed that are not data sets, formulas, analysis objects or folders, you have to specify a file name extension :

```
\Analysis\Document.doc
```

You can access the activated subfolder of a folder using the ActivatedFolder keyword, which does not have a '\' at the end:

```
\Analysis\ActivatedFolder
```

**Relative path names** provide the path to the target object from the folder from which you are trying to access it. Here, '..\' describes the folder from which you are trying to access it, and '..\' describes the parent folder, which is the folder above it in the folder

hierarchy. The following example is accessing a data set located in the neighboring folder called "Measurement2":

```
..\Measurement2\DataSet
```

This reads the folder name from which access is occurring:

```
.\.Name
```

You can access a data set located in the "Measurement1" subfolder as follows:

```
Measurement1\DataSet
```

To access an object located in the same folder, you only need to provide its name:

```
DataSet
```

or

```
Worksheet.wks
```

Relative path names never start with a '\', but instead either start with '.\' or '..\' or with the name of the subfolder.

---

**Note:** If you use an absolute path name or a relative path name that starts with '.\', then any activated subfolder will be ignored, which means that the path name always describes the specified object exactly, even if there is interference by an object with the same name in the activated subfolder.

---

## Indirect Access

The Indirection Operator (\$\$) gives you the option of accessing objects using a string with the object name. The following example shows a common use:

```
Name= InputText("Please enter the name of the data set")
Integral($Name$)
```

## Object Reference

An object reference is a reference to an object in FlexPro, e.g. to a formula or folder. You can use this type of object reference to access the value of the object reference or to access the properties of the object, e.g. access its name or comments. If you use the path name to a data object in FPScript, it will represent the data object value and will not provide an object reference. To construct it, you need to use the Set statement or the As Object keyword :

```
Dim Obj = DataSet As Object
```

or

```
Set Obj = DataSet
```

However, the Indirection Operator (\$\$) also provides an object reference for data objects:

```
Dim Obj = $"DataSet"$
```

is equivalent to:

```
Dim Obj = DataSet As Object
```

If a path name does not refer to a data object but instead refers to a folder, for instance, then in this case the folder represents the object reference. The following statements are therefore equivalent:

```
Dim Obj = Folder.fld As Object
```

or

```
Dim Obj = Folder.fld
```

FlexPro analyzes an object reference automatically if the value is required for additional calculations. You can use the Value Operator to analyze an object reference explicitly:

```
Dim Val = Value $"DataSet"$
```

### Comments

It is important that you add comments to your formulas, particularly for more complex programs. In FPScript, **comments** start with two forward slashes (/). A comment started this way extends to the end of the line. You can also use empty lines and lines containing only comments to structure your programs. Longer comments extending over several lines start with '/\*' and end with '\*/'.

### Statements

You can structure the FPScript code for your formula in one or more statements. Normally, you write every statement on its own line. You can, however, also write several statements on the same line. These have to be separated with a semicolon ';'. You can also split a statement over several lines. For lines which are to be merged with the lines following, you have to enter a backslash '\' or an underscore character '\_' as the final character.

### Assignment

You can use an assignment to assign different content to a variable. In a variable containing, for instance, a data series, you can specifically overwrite individual values or sections using the Indexed Assignment. With the Set statement or the As Object keyword you can assign an object reference to a variable. Afterwards, you can use this variable to access the properties of the referenced object.

## Making Decisions with If...Then...Else

The If...Then...Else statement is used to check whether a condition has the value TRUE or FALSE, in order to execute one or more statements conditionally. Normally, the condition is an expression with a comparison operator for comparing two variables or values. You can find information on comparison operators under "Comparison Operators". **If...Then...Else** statements can be nested to any depth.

## Executing a Block When a Condition Results in TRUE

In this case, the keyword **Else** is omitted:

```
If Value > 2 Then
    Value = 5
End
```

## Executing One Block When a Condition is TRUE and Another When the Condition is FALSE

The statements to be executed if the condition results in FALSE are inserted between the keywords **Else** and **End**.

```
If Value > 2 Then
    Value = 5
Else
    Value = 0
End
```

## Using Loops to Repeat Code

FPScript offers a host of operations to search for events and to process data sets in their entirety. In rare cases, however, you will have to use loops to run across the values in a data series, for instance. With a loop, you usually define a loop variable that enumerates the repetitions. This variable can then be used in an index operation, for instance, to access the corresponding value in a data set:

```
Dim Result, i                                // Declare variable
Result = DataSeries                          // Copy data series to be calculated
For Each Value i In DataSeries Do            // Run across all values in the data series
    If DataSeries[i] < 0 Then                 // Set values less than zero to zero in the result
        Result[i] = 0
    End
End
Return Result // Return the result data series
```

The above example uses a loop to find a negative value in a data series and to set it to zero. All statements between **Do** and **End** are executed several times, where the

variable `i` counts from 0 to the number of values in the data series of data series minus 1. The example also shows a conditional statement. The statement `Result[i] = 0.` is only carried out if the `i`th value in **DataSeries** is less than zero. This example also shows how **Indents** can make FPScript programs easier to read. All statements within the loop were indented by one tab position. The same applies to the conditional statement.

---

**Note:** You should only use loops if no appropriate function exists and if the problem cannot be described with an arithmetical expression. The integrated functions of FlexPro are many times faster than any loops programmed using FPScript. The example above can, for instance, be re-written as follows:

---

```
Dim Result, Idx          // Declare variable
Result = DataSet         // Copy data series to be calculated
// Determine position of all values < 0
Idx = ValuesAboveLevel(DataSeries, 0, EVENT_INDEX + EVENT_COMPLEMENT)
// Set values at these positions in the result to zero
Result[Idx] = 0
Return Result            // Return the result data series
```

FPScript offers you the following loops:

**While...End:** Executes a loop while the condition results in the value **TRUE**.

**Do...While:** Executes a loop until a condition results in the value **FALSE**.

**For...End:** Uses a counter to repeat the statements as often as is specified by the counter.

**For Each Value...End:** Traverses all values of a data set. No counter is used.

**For Each Row...End:** Uses a counter to iterate all of the values in a data series or a data matrix.

**For Each Element...End:** Uses a counter to count all of the items in a list.

**For Each Column...End:** Uses a counter to iterate all of the columns in a data matrix.

### Repeating Statements While a Condition is **TRUE**

The **While...End** loop first tests the condition and then executes the statement as long as the condition results in **TRUE**. The following example searches for the position in **DataSet**, where the first positive value occurs.

```
i = 0
While DataSet[i] < 0 Do
    i = i + 1
End
```



### Repeating Statements Until a Condition is FALSE

The Do...While loop first executes the statement one time and then tests the condition. The following example calculates random numbers until a number greater than or equal to zero is generated.

```
Do
    Z = Noise(0)
While Z < 0
```

### Using "For...End"

You can use For...End statements to repeat a block of statements as often as is specified. In loops, use a counter variable with a value that is increased each time the loop is executed.

```
a = "" # 3
For i = 0 To 2 End
    a[i] = TextInput("Enter text here")
End
```

### Using "For Each Row...End"

If, as in the example above, you want to iterate across all elements of a data set, then you can implement this more elegantly with the For Each Row...End loop.

```
a = "" # Input("Count")
For Each Row i In a Do
    a[i] = TextInput("Enter text here")
End
```

### Variables

In FPScript, you can use variables for interim results. A variable acts as a placeholder where you can store any results of calculations and recall them when necessary. The name of the variable allows access within the formula to the storage location to which it has been assigned.

Before using a variable for the first time, the variable must be declared using the Dim Statement. This creates storage for one or more variables. A variable, which has not yet been assigned a value, is of the Empty data type. You can use an Assignment to assign a value to it. In a variable containing, for instance, a data series, you can specifically overwrite individual values or sections using the Indexed Assignment.

The range of validity for a variable is restricted to the formula in which you use it. Thus,

you can use the same variable name in various formulas without risking conflicts.

### Example

```
Dim i Series          // Declares the two variables i and Series.
i = 1n                // Initializes a variable i with the integer scalar value 1.
Series = 0. # 100n    // Creates a data series with 100 zeros.
```

For complex calculations, it often makes sense to split up the calculation into several statements. In a first statement you could calculate, for instance, a partial expression that you then use in further statements. To do this, it is important to assign the results of the calculations to variables so you can use them later:

```
Dim Angle              // Declare variable
Angle = 2 Pi * f * t    // Calculation of angle
2 * Sin(Angle) + 3 * Cos(Angle) // Calculation of an overlapped oscillation
```

Here, the calculation of the angle is performed on a separate line.

### Building Data Structures

FPScript provides operators and functions to help you build your own data structures.

#### Linear Data Series

For instance, this Data Series operator creates a data series with ascending values:

```
(100, 0., 1.)
```

**100** is the number of values, **0.** is the starting value, and **1.** is the increment. If you prefer to specify the starting value, the end value and the sampling interval, use the Series Function:

```
Series(0, 100, 1)
```

generates a data series with 101 values with a starting value of **0**, end value of **100** and sampling interval of **1**. Here, the data type is always a 64-bit floating point value.

If you use quantities, you can specify different units, but they must be compatible:

```
(2000, 0 s, 1 ms) OR Series(0 s, 2 s, 1 ms)
```

The result contains the unit that is furthest to the left and thus is equal to:

```
(2000, 0 s, 0.001 s)
```

### Data Series and Matrices with Constant Content

The Multiply operator is used to create data series or data matrices with constant content:

```
1.5 # 100n
```

The above example shows a data series with **100** times the value of **1.5**. The scalar value 1.5 has been multiplied 100 times.

`1.5 # 100n # 20n` generates a two-dimensional data matrix with **20** data series of **100** values each.

`TRUE # Shape(Matrix)` generates a matrix of Boolean values whose number of columns and rows corresponds to an existing matrix.

Even here you can multiply quantities:

```
1.5 N m # 100n
```

### Data Series and Matrices with Variable Content

You can also bundle different scalar values into data series and data series into data matrices:

```
{ 1.5, 2.6, 7.8 }
```

The Bundle operator bundles the scalar values into a data series with three values. The length of the list is not restricted. Of course, in addition to constant scalar values, you can write arithmetical expressions that return scalar values.

`{ DataSeries1, DataSeries2, DataSeries3 }` bundles three one-dimensional data series into one two-dimensional data matrix. The data series should be the same length.

All elements of a data series or data matrix are of the same data type. This does not apply to components of aggregate data structures; for instance, the X component of a signal can have a different data type than the Y component.

Instead of numeric values, you can also bundle quantities. It is sufficient to specify a unit for the first value; this will then apply to the other values.

```
{ 1.5 kg, 2.6, 7.8 }
```

If you specify different units, they must be compatible. The result contains the unit of the first quantity:

```
{ 50 mV, 0.5 V, -0.7 V }
```

 corresponds to the input of `{ 50 mV, 500 mV, -700 mV }`.

### Lists with Multiple Independent Elements

The List operator brings together several values or quantities that, for their part, can

have any data structures and data types. Each element of a list has the option of having a name. Since each list element can in turn also be a list, any deeply nested structures can be built.

[ <Solution> Data Series, <ChiSquare> Scalar Value ] compiles the solution vector from a curve fitting and the goodness-of-fit measure into a list. The names provided in angle brackets are optional and describe the list elements.

**Note:** The use of list element names is no longer recommended in FlexPro version 10 and higher. Instead, use the AssignHeader function to assign header information to the list elements.

Lists are used for multichannel analyses or when a single calculation produces several independent results that cannot be calculated separately.

To access a list element, you can use its name, if available, or the position of the element in the list:

List.ChiSquare accesses the second element in the list above.

List.[1n] gives you the same result without using the name of the element.

You can use the ListToSeries function to convert a list with items of the same structure into a data series, data matrix or signal series.

### Constructing Aggregate Data Structures

You can use the Signal function to create signal, signal series and space curve aggregate data structures from the basic data series and data matrix data structures:

Signal(Amplitude, Time) joins two data series with measurement values and the corresponding time values to form a signal. **Amplitude** and **Time** are then the components of the signal.

### Concatenating Data

The Concatenation operator appends data sets to each other:

DataSeries1 : DataSeries2

You can also append signals to each other. In this case, the X component of the second signal is shifted automatically so that it follows seamlessly after the first.

If you concatenate quantities, the unit of the argument on the right is adjusted to the one on the left.

## Void Values

32- and 64-bit floating point values can have the value "void". These types of void values can represent, for instance, outliers in a data series.

Arithmetical operations, such as addition or multiplication with void values provide a void result.

All comparison operations return FALSE if one of the operands is void. The 'is equal to', 'is greater than or equal to' and 'is less than or equal to' comparison operators, however, return TRUE if both operands are void.

With the Analysis of data sets containing void values, these void values are ignored as much as possible. For example, in the case of the formation of a mean value, these values are not taken into consideration.

Due to their structure, some algorithms have the property of dispersing the void attribute. For example, if you calculate an FFT of a time signal with void values, then the result is practically completely void. You should not apply these types of procedures to data sets with void values.

You can display the value "Invalid" in formulas or when editing data sets via a ?.

Constant	Value
?s	Void 32-bit floating point value
? or ?L	Void 64-bit floating point value
? V	Void 64-bit floating point quantity with the unit V
'?.?.?' or '?/?/?'	Void calendar time value
'?:?:?' or '?:?:?:?'	Void time span value

You can use Cursors to make the values under the active cursor or the entire range between the cursors void. For example, you can use this option to mark outliers in a measured signal as void.

You can use the SignalCorrection analysis object\* to remove void values from data sets or to interpolate them.

The functions SearchVoidValues \*, ValuesInInterval \* can be used to search for void values in a data set.

\* Not available in FlexPro View.

## Constants

Constants used in FPScript are subject to specific syntax rules, where, based on the notation used, FlexPro can recognize the data type as well as the value.

**Integers** can be written in decimal, octal and hexadecimal notation. Octal numbers are written with a leading zero and can only contain the numbers 0 to 7. Hexadecimal numbers are written with a leading combination 0x and can contain numbers 0 through 9 and A through F. Displaying hexadecimal and octals is interesting for digital data because there is a direct relationship between the numbers and the bits that represent them. For octal numbers, three bits are the equivalent of one digit, and with hexadecimal numbers, four bits are the equivalent of one digit. Please note that the only notation is different for integers. Internally, all integers are saved in binary form.

FlexPro can process floating point values with 16-bit, 32-bit and 64-bit resolution. You can control the resolution by appending an 'S' or 's' for "short" or an 'L' or 'l' for "long". If you append the suffix 'N' or 'n', then this corresponds to 32-bit or 64-bit integers, depending on which format can store the constant.

Examples of 16-bit integers are:

123s	Decimal
0427s	Octal
0x1afs	Hexadecimal

Examples of 32-bit integers are:

100000n, 12n	Decimal
01024341121, 0123	Octal
0x10A00FF, 0x1A, 0x1a	Hexadecimal

Examples of 64-bit integers are:

10000000000n, 12L	Decimal
071024341121, 0123L	Octal
0x1010A00FF, 0x1AL, 0x1aL	Hexadecimal

**Floating point values** consist of a mantissa with whole and fractional digits and an optional exponent with a leading e or E. Floating point values can also be a void value. This value is displayed as ?.

FlexPro can process floating point values with 32-bit and 64-bit resolution. 32-bit floating point values are accurate to approximately seven decimal places, and 64-bit floating point values are accurate to approximately 15 decimal places. You can control the resolution here as well by appending an 'S' or 's' for "short" or an 'L' or 'l' for "long". If

you do not append a suffix, then this always corresponds to 64-bit floating point values, which means that the suffix 'L' can always be left off.

---

**Note:** For the FPScript code, the decimal point (.) is always the decimal symbol, even if in the Regional and Language Options settings of the Windows Control Panel you have specified the comma (,), which is commonly used in some European countries.

---

Examples of 32-bit floating point values are:

```
1.23s, 1.234e-12s, 1e+13s, 0.1e12s, 5.s, ?s
```

FlexPro does not recognize whether the range of values or the resolution has been exceeded. If you would like to obtain 64-bit floating point values, you need to omit the 's'.

Examples of 64-bit floating point values are:

```
1.23, 1.234e-12, 1e+13, 0.1e12L, 5.L, ?, ?L
```

In general, spaces may not be used within numbers, not even between mantissa and exponent. If an overflow occurs during a calculation, FlexPro cannot recognize this! You have to ensure that the range of values of the data type is sufficient to perform the calculation.

Example:

```
1000s * 1000s    Generates an overflow because the result, 1,000,000, does not fit into
                  a 16-bit integer! (Solution: 1000n * 1000n)
```

---

**Note:** Interpreted like decimal numbers to which you have not added a suffix or decimal point, depending on whether the setting **Interpret decimal numbers without a decimal point as floating point values** is enabled on the **General** tab of the database's properties dialog box. This option is enabled by default, which means that the numbers are interpreted as 64-bit floating point values. This setting prevents accidental rounding errors that may occur when using integer arithmetic. Integral division 1 / 2 results in 0 and not 0.5, for instance.

---

**Complex numbers** consist of a real and imaginary part, either in the form of a + bi or separated by commas and placed in parentheses. You can use all of the above notations for constants for the real part and the imaginary part. If this results in different data types for the real and imaginary parts, then these are adapted to each other before being combined into a complex number so that there is no loss of resolution. **Within the FPScript, the comma (,) is always the list separator**, even if in the Regional and Language Options settings of the Windows Control Panel you specified the

semicolon (;), which is commonly used in some European countries.

Examples of complex numbers are:

(1.23s, 1.234e-12s) , (1e+13, 0x1), (5.L, ?), 1 + 2i, -3.2 - 7.6i, 5.2i, ?i, 2s + 5si

**Strings** are entered as text set off by quotation marks. There are specific control sequences available for special characters, such as a tab or a return character. These sequences all start with a '\ ' character.

---

**Note:** If you would like to use the '\ ' character as part of your text, you will have to include it in the string twice. The file path C:\Data.txt must be written as the string "C:\\Data.txt" .

---

Control sequence	Stands for
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\'	Single quotation mark
\"	Double quotation mark
\\	Backslash
\ooooo	UNICODE characters in octal notation
\xhhhh	UNICODE characters in hexadecimal notation

Examples of strings are:

"A Text", "C:\\Data\\File.txt",", "Two\r\nlines"

The **Empty** data type can also be used as a constant to empty a variable used in a formula, for instance. To do this, just write Empty :

Result = Empty

**Calendar time values** consist of the date and time, with seconds in fractional digits being optional. Only two formats are allowed within FPScript. If you separate the date elements with '.', then the sequence will be Day.Month.Year, and if you use '/' as the separator, then this is interpreted as Month/Day/Year. If you use '-' as the separator, then it is interpreted as Year-Month-Day. The year can be entered as two or four digits. In the two-digit format, for values smaller than 70, 2000 is added, otherwise 1900 is



added. For floating point data types, as well as for calendar times, the value is void if '??/?' appears. The elements for time are always specified in the format HH:MM:SS.ss. You can omit time elements starting from the right if they are zero. Constants in calendar time format must generally be written using single quotation marks:

'23.10.2004 12:13:24.123'	Calendar time value with a resolution of 1ms
'10/23/2011 12:13:24.123'	Calendar time value written in English notation with a resolution of 1ms
'2004-10-23 12:13:24.123'	Calendar time value arranged as Year-Month-Day
'23.10.2004 12'	Calendar time value 23.10.2004 at twelve noon
'23.10.2004'	Calendar time value 23.10.2004 at midnight
'1.1.24'	Calendar time value 1.1.2024 at midnight
'1.1.98'	Calendar time value 1.1.1998 at midnight
'?/?/?'	Invalid calendar time value

---

**Note:** The calendar time value constants are interpreted in local time, i.e. these are converted into the UTC format used for internal storage taking into account the time zone settings in Windows and, if applicable, daylight savings time.

---

**Timespan values** consist of the elements day, hour, minute, second. You can specify seconds using fractional digits as an option. The elements are specified in the format D:HH:MM:SS.ss. You do not have to specify all elements. If you specify two elements, then this is interpreted as HH:MM. If you specify three elements, then they are interpreted as HH:MM:SS, and four elements are then interpreted as D:HH:MM:SS. For output, FlexPro uses three or four elements as the standard. The day is omitted if the timespan is less than 24 hours. Constants in calendar time format must generally be written using single quotation marks:

'12:13:24.123'	Timespan value with a resolution of 1 ms
'13:24'	Timespan value of 13 hours and 24

	minutes
'2:01:30:00' or '49:30'	Timespan value of 2 days, 1 hour and 30 minutes
'?:?:?'	Invalid timespan value

**Note:** The calendar time and timespan values have second as the implicit unit. The unit symbol, however, is not displayed.

Pre-defined constants

FPScript recognizes a series of pre-defined constants. The names of the constants cannot be used as object names or as variable names. These are mathematical constants  $\text{PI} = 3.141592654\text{L}$  and  $\text{E} = 2.718281828\text{L}$  as well as a series of constants that have pre-defined meaning as the parameters for functions.

Operators

In FPScript you can use a variety of operators, e.g. for arithmetical and logical operations, comparison and concatenation operations. You can use the operators to calculate individual values or entire data sets, data matrices, signals, signal series or space curves.

Operator Precedence

If several operators occur in an expression, each part is evaluated and resolved in a predefined order. This order is described as operator precedence. You can override the operator precedence using parentheses to force the evaluation of parts of an expression before other components. Operations in parentheses are always executed before the operations that are not in parentheses. However, standard operator precedence also applies within the parentheses.

If an expression contains operators of different categories, then these are evaluated in the order specified below. All comparison operators have the same priority, i.e. they are evaluated from left to right in their order of appearance. Arithmetical and logical operators are evaluated in the following order:

Description	Symbol
Conversion	
Convert to Boolean Value	Boolean Value

Description	Symbol
Convert to 16-Bit Integer	Integer16
Convert to 32-Bit Integer	Integer32
Convert to 64-Bit Integer	Integer64
Convert to 32-Bit Floating Point Value	FloatingPoint32
Convert to 64-Bit Floating Point Value	FloatingPoint64
Convert to Complex 16-Bit Integer	ComplexInteger16
Convert to Complex 32-Bit Integer	ComplexInteger32
Convert to Complex 64-Bit Integer	ComplexInteger64
Convert to Complex 32-Bit Floating Point Value	ComplexFloatingPoint32
Convert to Complex 64-Bit Floating Point Value	ComplexFloatingPoint64
Convert to String	String
Convert to Calendar Time	CalendarTime
Convert to Time Span	TimeSpan
Convert unit	Unit
Convert quantity to value	Value
<b>Arithmetic</b>	
Unary Negation	-
Exponentiation	^
Multiplication	*
Division	/
Division Remainder	%
Addition	+
Subtraction	-
Operational Sign	#
<b>Complex Numbers</b>	
Form Real Component	Real
Form Imaginary Component	Imag
Form Conjugate-Complex Number	*
<b>Comparison *</b>	
Less Than	<
Greater Than	>

Description	Symbol
Less Than or Equal To	$\leq$
Greater Than or Equal To	$\geq$
Equality	$==$
Inequality	$\neq$
<b>Logical</b>	
Binary Negation	$\sim$
Binary Conjunction	$\&$
Binary Exclusion	$\text{XOr}$
Binary Disjunction	$ $
Logical Negation *	Not
Logical Conjunction *	And
Logical Disjunction *	Or
<b>Data Organization</b>	
Concatenation	:
Multiply	#
Form Complex Number	( , )
Bundle	{ , }
Form Data Series	( , , )
Form List	[ , ]
<b>Data Access</b>	
Component	.X.Y.Z
Property	.
List Element	.[ ]
Index	[ , ]
Value Index	[[ , ]]
Indirection *	\$\$

If multiplication and division occur in an expression, each operation is executed from left to right in the order of appearance. The same procedure applies when addition and subtraction or concatenation and multiplying occur in an expression.

## Associativity

The exponentiation operator is the only operator that is right-associative. This means that several exponentiations are processed in sequence from right to left. All other operators are processed from left to right.

## Processing Logical Expressions

A logical conjunction (Boolean AND operation), or sequences from it are only evaluated until the result is defined, i.e. until the first expression results in FALSE. The same applies to the logical disjunction (OR operation). These are evaluated until the first expression results in TRUE. This does not apply to the bitwise logical operations.

In the following example, `Test(DataSet2)` is only called when `Test(DataSet1)` results in true. If `Test(DataSet1)` results in FALSE, then the final result of the Boolean AND operation is defined as FALSE.

```
If Test(DataSet1) And Test(DataSet2) Then
    ...
End
```

\* This operator and all operators of this category are not available in FlexPro View.

## Legacy Functions in FPScript

The number of available FPScript functions is constantly being expanded and reorganized. Consequently, FPScript functions also become outdated because their functionality ends up being covered by other functions. For the purpose of compatibility, these legacy functions are still supported by FlexPro, but are no longer offered as an option in the Function Wizard and are also no longer mentioned in the documentation. These functions are marked by FlexPro with a '\_' sign before the name of the function.

The following is a list of all legacy functions and their successors:

Function	Last Version Used	Replaced By	Considerations
<code>_BartlettWindow</code>	FlexPro 6	<code>DataWindow(s, WIN_BARTLETT)</code>	
<code>_BlackmanWindow</code>	FlexPro 6	<code>DataWindow(s, WIN_COS3BLACKMAN)</code>	
<code>_Bursts</code>	FlexPro 7	Bursts	Algorithm has been changed.
<code>_ConnesWindow</code>	FlexPro 6	<code>DataWindow(s, WIN_BISQUARE)</code>	

Function	Last Version Used	Replaced By	Considerations
_CosineWindow	FlexPro 6	DataWindow(s, WIN_SINE)	FFtn is no longer limited to values based on the power of 2. Please note, however, that FFtn provides a non-normalized complex magnitude spectrum, while FFT calculates complex amplitudes.
_FFT	FlexPro 6	FFtn(s) / NumberOfRows(s) / 2 FourierSpectrum(s, SPECTRUM_AMPLITUDE, WIN_RECTANGULAR)	
_FlatTopWindow	FlexPro 6	DataWindow(s, WIN_FLATTOP)	
_HammingWindow	FlexPro 6	DataWindow(s, WIN_HAMMING)	
_HanningWindow	FlexPro 6	DataWindow(s, WIN_HANNING)	IRFFT expects an amplitude spectrum and IRFFtn expects a magnitude spectrum as the argument.
_IIRBandFilter	FlexPro 7.0.13	IIRFilter and Filter	
_IIRFilter	FlexPro 7.0.13	IIRFilter and Filter	
_IRFFT	FlexPro 6	IRFFtn	
_KaiserBesselWindow	FlexPro 6	DataWindow(s, WIN_KAISERBESSEL)	Algorithm has been changed.
_LevelCrossing	FlexPro 4	LevelCrossings	
_LevelCrossings	FlexPro 7	LevelCrossings	
_LocalExtrema	FlexPro 4	Extrema	
_LocalMaxima	FlexPro 4	Extrema	Algorithm has been changed.
_LocalMinima	FlexPro 4	Extrema	
_NegativePeaks	FlexPro 4	Extrema	
_NextBurst	FlexPro 7	NextBurst	
_NextLevelCrossing	FlexPro 7	NextLevelCrossing	Algorithm has been changed.
_NextSlope	FlexPro 7	NextSlope	Algorithm has been changed.
_ParSplineX	FlexPro 3	ParametricSpline	
_ParSplineY	FlexPro 3	ParametricSpline	
_PositivePeaks	FlexPro 4	Extrema	

Function	Last Version Used	Replaced By	Considerations
_Random	FlexPro 6	Noise	
_Rectangle	FlexPro 6	Square	
_ReadASCII	FlexPro 3	ReadTextFile	
_ReadORP_ORMFile X	FlexPro 5	ReadORP_ORMFile	
_ReadORP_ORMFile Y	FlexPro 5	ReadORP_ORMFile	
_SigMax	FlexPro 3	Maximum	
_SigMin	FlexPro 3	Minimum	
_Slopes	FlexPro 7	Slopes	Algorithm has been changed.
_Triangle	FlexPro 6	Sawtooth	
_WelchWindow	FlexPro 6	DataWindow(s, WIN_WELCH)	
_WindowFFT	FlexPro 6	STFTSpectrum(s, SPECTRUM_AMPLITUDE, .. )	
_ZeroCrossings	FlexPro 4	LevelCrossings	

### Accessing Header Information

FlexPro assigns a data object Header Information reference to an FPScript value that originates from this data object, e.g. from a formula. You can therefore access the data object attribute via the value. The header information information is the properties of the ValueObject object, including the parameter list. If an FPScript value is passed as an argument or as a return value from a value in a different formula, then the value can be used to access the header information without an object reference.

The following code uses the variable x to access the X component comments of the data set containing the x value:

```
Dim x = \Data\DataSet
```

```
x.CommentsX
```

You can also have write-access to an attribute:

```
x.CommentsX = "Time"
```

```
x.UpperRangeLimitX = 1.3
```

Write access in this case is not to the data object upon which it is based (\Data\DataSet

in the example above). Instead, `x` is assigned a local copy for the property to be changed. This copy is then set to the assigned value. You can delete the local copy by deleting the associated property:

```
x.CommentsX = ""  
x.UpperRangeLimitX = ?
```

The associated attributes will then be removed from the data set at the next read access attempt.

### Parameters

The complete parameter list is always provided as a local copy. If when using write access a name is specified that does not have any parameters, the name will be created automatically.

```
x.Parameters("MyParam") = 1.3
```

Apply the Empty value to delete a parameter again:

```
x.Parameters("MyParam") = Empty
```

You can verify whether a parameter is in the list using the Index property. It will return 0 if the parameter does not exist:

```
x.Parameters("MyParam").Index
```

### Lists

Forms that provide a list with multiple data sets as the result are a special feature. If the formula bundles only data sets into a list, the list elements obtain the header information of the data sets from which they originate:

```
Dim x = [Signal1, Signal2]  
x.[0].Name returns "Signal1"
```

If the formula calculates a result and the result is returned as a list, you will have to assign the header information to the result explicitly using the `AssignHeader` function. This function assigns the formula header information to each list element and sets the property `Name` on the element names defined on the formula **Results** tab.

```
Dim Data = [Signal1, Signal2]  
Dim Result = Integral(Data)  
Dim x = AssignHeader(Result, this, Data)
```

`x.[0].Name` returns "Signal1Integral", if on formula the **Results** tab, the option **Automatic** is selected and "Integral" has been entered as the template.

`x.[0].CommentsY` returns the Y comments of the formula containing the FPScript code (not the Y comments of the Signal1 data set).



If the list elements are copies of other data objects, they will obtain the header information of the data objects from which they originate:

```
Dim List = [Signal1, Signal2]
List.[0].Name returns "Signal1"
```

## Components

If an FPScript variable contains a component of an aggregate data set, then the component is used as the template for the properties for which a component can be specified optionally as the argument if you omit the argument:

```
Dim x = DataSet.X
x.Quantity corresponds to x.Quantity(fpDataComponentX)
```

It is therefore easy to access the attributes of each relevant component.

## 5.4 FPScript Functions

FPScript offers you the option of programming your own functions in FPScript. You can create these as you would regular formulas; however, you need to use the Arguments statement in the first code line to declare arguments for the function. This turns the formula into a function which must be supplied with arguments when called. You can also save these types of functions in template databases, thus expanding the number of FPScript functions.

For example, a function called `sum` would appear as follows:

```
Arguments a, b
a + b
```

This function can now be called in a different formula:

```
Sum(1 V, 2 mV)
```

or

```
Sum(DataSet1, DataSet2)
```

The arguments *a* and *b* serve as formal parameters to represent the arguments passed. You can use these just like local Variables.

You can omit any items in the argument list when calling the function. These items then have an empty data type:

```
Sum( , 2) is therefore the equivalent to Sum(Empty, 2) and Sum(1) is equivalent to Sum(1, Empty).
```

Please note that in the function *Sum* no Return statement is necessary. The result of the last statement in a formula is automatically used as the result of the formula.

### Saving FPScript Functions as a Template

If you save an FPScript function in a template database, you can use it in your project databases as you would a built-in function. To save a function, use a wizard in which you can specify a comment for each argument and define the permitted data types and structures. FPScript functions saved this way

- are displayed in the wizard for inserting a function into a formula in the **Customized** category,
- are supported by the FPScript Editor's Assistance feature,
- support optional arguments with default values, and
- when called, automatically check arguments passed for permitted data types and data structures.

### Sharing FPScript Functions

FlexPro Professional and FlexPro Developer Suite support shared template databases. You can share FPScript functions that you save in this type of template database with your colleagues. Use the shared template databases to create function libraries for your department or company.

### Working with FPScript Functions

#### Creating an FPScript Function

1. In the Folders window, select the folder in which you want to place the new FPScript function.
2. Select the command **Insert Formula** from the **FlexPro Objects** toolbar.

3. Enter the desired name in the Object List input field, e.g. **Normalize**.
4. Enter the function code in the FPScript Editor. In the first line, use the **Arguments** statement  
Arguments Data  
Data / Maximum(Data)

### **Saving an FPScript Function as a Template**

1. Select the FPScript function from the Object List, right-click with the mouse and choose **Save as Template** from the context menu.
2. Follow the steps in the Template Wizard and use the dialog box Help if you require additional assistance.

### **Calling an FPScript Function**

- Use an FPScript function as you would a built-in function by entering the name, followed by the argument list, in the FPScript code, as in the following example:  
Normalize(Signal)
- If the function to be called was not saved in a template database and is also not to be found in the same folder as the formula from which you want to call the function, you have to specify the path name, such as follows:  
\MyFunctions\Normalize(Signal)

### **Applying FPScript Functions to Data Sets**

Often, identical calculations have to be performed for a variety of data sets. One way of automating this process is to save the data sets in different subfolders and to activate the folder from which the data for an analysis is to be taken.

The second option, which is described here, is to pack the calculation rule into an FPScript function and then to use this as a template for generating new formulas.

FPScript offers you the option of programming formulas with arguments. You can then use these formulas like functions. A **Normalized** formula, which normalizes your argument to a range of values from 0 to 1, looks like this, for example:

```
Arguments Data  
Data / Maximum(Data)
```

You can now use this function in other formulas. For instance, you can add a **SignalNormalized** formula containing the following code:

```
Normalized(Signal)
```

It is very easy to create these types of formulas with FlexPro: You only have to click on the data sets to which you would like to apply the function, use your mouse to select

them from the Object List and drag them onto the function in the object list. For each data set, FlexPro then adds a new formula that applies the function to this data set. In the example above, drag the data set **Signal** to the function **Normalized** in the Object List. FlexPro will then automatically add a new formula named **SignalNormalized**, which contains the code specified above. You can then use this new formula for further calculations or for displaying in documents.

You can also use functions with several arguments. In this case, the first argument used is the data set that you dragged onto the function with the mouse. You can specify additional arguments in a dialog box.

## 5.5 Analysis Objects

FlexPro offers you special analysis objects for all common analyses. These are FScript formulas that FlexPro automatically creates. You can set the parameters for the analysis object using its Properties dialog box. The formula will then be adjusted automatically. Analysis objects thus offer automated analyses without any programming.

In addition to simple parameterization, many analysis objects can automatically generate the physical units and comments for their results. You can transform analysis objects into data sets or formulas at any time.

In the Reference section, you will find an alphabetical list of the available analysis objects.

---

**Note:** In FlexPro View only a small set of analysis objects for basic statistics is available.

---

### Analyzing several data sets

For multichannel analyses, the same calculations are used for multiple channels. In this case, FlexPro can create one analysis object for all output data sets or an analysis object for each output dataset.

If you create only one analysis object, it will provide all calculated data sets as a list. The advantage of this is that you only have to parameterize one object and an analysis with a better structure. This option is also particularly useful if all channels need to be processed in the same format, such as when filtering, integrating and displaying 16 channels.

If you decide to create one analysis object per output data set, you can parameterize the each analysis individually, and it will be easier to select individual results for further processing.

## Working with Analysis Objects

### Creating an Analysis Object

#### To create an analysis for a data set or a range from a data set:

1. In the Folders window, select the folder where you want the new analysis object to be placed.
2. Select the data to which the analysis is to be applied. You can select a data set, formula or cell range in the data view or data set window. This step is optional. If you do not select any data, you have to specify the data later in the Properties dialog box of the analysis object.
3. Click on **New** in the menu bar, point to a category and then select the desired analysis procedure in the submenu that appears.
4. In the **Properties** dialog box that appears, set the parameters for the new analysis object.

---

**Note:** Some analysis procedures require more than one argument. In this case, you have to select multiple data sets in the Object List or specify additional data sets Properties dialog box of the analysis object.

---

#### To create the same analysis for multiple data sets:

1. In the Folders window, select the folder in which you want to place the new analysis objects.
2. Select the data sets or formulas to which the analysis is to be applied.
3. Click on **New** in the menu bar, point to a category and then from the submenu that appears select an analysis procedure that only calculates a data set.
4. In the **Analyzing several data sets** dialog that appears, select whether you want to create one analysis object that calculates all data sets or an analysis object for each data set. Use the Help in the dialog box to learn more about the advantages of each option.

---

**Note:** You can also choose the option you want without viewing the dialog box: Hold down the CTRL and SHIFT keys while you select the analysis from the **New** menu to create an analysis object for each data set. Alternatively, press and hold the CTRL key to create a single analysis object.

---

5. If you created only one analysis object, a Properties window where you can set the

parameters for it will appear. Otherwise, the new analysis objects will be selected in the Object List. You can set parameters for all objects at once in the FlexPro Properties window.

---

**Note:** Not all analysis objects are available in FlexPro View.

---

**Editing an Analysis Object**

**Editing a Single Analysis Object**

- 1. Double-click on the object in the Object List to display its Properties dialog box.
- 2. On the **Options** tab you can adjust all settings for the analysis procedure.
- 3. Specify the data sets you want to analyze on the **Data** tab.

**Editing Several Similar Analysis Objects**

- 1. Select the analysis object that you would like to edit from the Object List.
- 2. Adjust the desired settings in the FlexPro Properties window.

**5.6 Reference**

**Analysis Objects and Templates**

**Signal Analysis**

Analysis object	Available in	Used for
Convolution	Standard, Professional, Developer Suite	Calculates the convolution product of two data sets.
Whole-Body Vibration	Human Body Vibrations Option	Used to analyze mechanical vibrations affecting the human body.
Hand-Transmitted Vibration	Human Body Vibrations Option	Used to analyze hand-transmitted vibrations that are transferred from vibrating tools through the palms and fingers to the hands and arms.
Correlation	Standard, Professional, Developer Suite	Calculates either the autocorrelation function of a data set or the cross-correlation of two data sets.

Signal Sampling	Standard, Professional, Developer Suite	Reduces or increases the number of values of a data set using different methods or samples it for a new X data series.
Signal Analysis	Standard, Professional, Developer Suite	Computes the derivative or integral of a data set or calculates the area under the curve or the signal energy of a data set.
Signal Scaling	Standard, Professional, Developer Suite	Transforms a data set linearly or using a characteristic curve.

## Spectral Analysis

Analysis object	Available in	Used for
Family of Acceleration Hyperolas	Order Tracking Option	Calculates a family of acceleration hyperbolas that can be displayed in a 3D waterfall diagram, for instance.
Cepstral Analysis	Spectral Analysis Option	Non-linear Fourier method for the deconvolution of two signals.
Fourier Spectral Analysis	Standard, Professional, Developer Suite	Provides three different spectral procedures.
Uneven Data Fourier Spectral Analysis Object	Spectral Analysis Option	Generates a Lomb-Scargle periodogram for data sets with unevenly spaced X values and for data containing void values.
Split Orders	Order Tracking Option	Splits the orders contained in an order tracking analysis for different speeds into separate objects.
Harmonic Estimation	Spectral Analysis Option	Generates a parametric model of the signal.
Spectral Estimator	Spectral Analysis Option	Provides three spectral estimators with high frequency resolution.
Fourier Cross Spectral Analysis	Spectral Analysis Option	Selection of four different Fourier cross-spectrum analyses: Cross-spectrum, cross-periodogram, coherence, transfer function
Order Analysis	Order Tracking Option	Calculates an order tracking analysis for speed-dependent oscillations.
Time-Frequency Spectral Analysis	Standard, Professional, Developer Suite	Provides two different time-frequency spectral procedures for non-stationary data.
Shock Response Spectrum	Spectral Analysis Option	Calculates the shock response spectrum from the responses of several SDOF systems that are

excited by an accelerometer signal.

## Filters

Analysis object	Available in	Used for
CFC Filter	Digital Filters Option	Filters signals with a CFC (Channel Frequency Class) filter.
Event Isolation	Standard, Professional, Developer Suite	Searches data sets for various events.
FIR Filter (Window Method)	Digital Filters Option	Filters signals or calculates the impulse response of a finite impulse response (FIR) filter using windowing.
FIR Filter (Equiripple Method)	Digital Filters Option	Filters signals or calculates the impulse response of a finite impulse response (FIR) filter using the equiripple method.
IIR Filter	Digital Filters Option	Filters signals or calculates the numerator and denominator coefficients of an infinite impulse response (IIR) filter.
Signal Smoothing	Standard, Professional, Developer Suite	Smooths a data set.
Signal Correction	Standard, Professional, Developer Suite	Deletes or interpolates void values from floating-point data sets.
Time Filter	Standard, Professional, Developer Suite	Filters periodic time segments from a data set with calendar time values.

## Curve Fitting

Analysis object	Available in	Used for
2D Approximation	Standard, Professional	Approximates a selectable model function $Y(X, Z)$ to 2D data based on the least square error method.
Approximation	Standard, Professional, Developer Suite	Approximates a selectable model function $Y(X)$ to data based on the least square error method.
CurveTransformation	Standard, Professional, Developer Suite	Converts space curves into signal series and vice versa.



Analysis object	Available in	Used for
Linear interpolation	Standard, Professional, Developer Suite	Carries out a linear interpolation for data.
Non-Linear Curve Fitting	Standard, Professional, Developer Suite	Fits a model function based on an independent variable and several parameters to a given data set.
Surface Interpolation	Standard, Professional, Developer Suite	Provides four methods for smoothing surfaces.
Parametric Spline Interpolation	Standard, Professional, Developer Suite	Carries out a parametric spline interpolation for data.
Peak Fitting	Standard, Professional, Developer Suite	Approximates multiple instances of a peak function to a pre-defined data set.
Regression	Standard, Professional, Developer Suite	Calculates various regressions for a data set.
Signal Sampling	Standard, Professional, Developer Suite	Provides different methods for increasing or reducing the number of values of data sets or samples them for new X values.
Spline Interpolation	Standard, Professional, Developer Suite	Carries out a spline interpolation for data.

## Statistics

Analysis object	Available in	Used for
ANOVA	Statistics Option	Performs a Fisher analysis of variance for a series of samples from a normally distributed population.
Goodness-of-Fit Test	Statistics Option	Performs goodness-of-fit tests for samples from a normally or exponentially distributed population.

Outlier Correction	Statistics Option	Deletes outliers from a sample.
Outlier Test	Statistics Option	Checks one sample from a normally distributed population for outliers.
Statistical Quantity	View, Standard, Professional, Developer Suite	Calculates various statistical quantities of a data set.
Variance Test	Statistics Option	Tests whether the variances from samples - originating from a normally distributed population - are significantly different or not.
Distribution	Statistics Option	Calculates the distribution or density function of the most important theoretical distributions.
Confidence Interval	Statistics Option	Calculates the confidence intervals for the parameters of the normal distribution of a population based on a sample.

### Counting Procedures

Analysis object	Available in	Used for
Histogram	Standard, Professional, Developer Suite	Creates histograms for data.
Counting	Count Option	Counts a signal with one of the procedures described in the DIN 45667 standard.
Count Matrix	Count Option	Determines the Rainflow matrix or the Markov (transition) matrix of a signal.
Rainflow Count	Count Option	Derives univariate collectives from an existing Rainflow matrix or Markov matrix.
Compound Count	Count Option	Counts two signals synchronously with some of the procedures described in the DIN 45667 standard.

### Acoustics

Analysis object	Available in	Used for
Loudness	Acoustics Option	Calculates the loudness from third octave spectra (ISO 532:B) or octave spectra (ISO 532:A).
Octave Analysis	Acoustics Option	Performs an octave analysis with digital filters in the time domain.
Sound Power	Acoustics Option	Determines the sound power according to international standards.

Analysis object	Available in	Used for
Sound Level	Acoustics Option	Determines sound levels from recorded microphone signals according to international standards.

### Other

Analysis object	Available in	Used for
Signal	View, Standard, Professional, Developer Suite	Calculates various synthetic signals.

## FPScript Operators

### Arithmetic

Operator	Available in	Used for
Addition (+)	View, Standard, Professional, Developer Suite	Adds two expressions.
Division (/)	View, Standard, Professional, Developer Suite	Divides two expressions.
Division Remainder (%)	View, Standard, Professional, Developer Suite	Determines the remainder of a division.
Multiplication (*)	View, Standard, Professional, Developer Suite	Multiplies two expressions.
Negation (-)	View, Standard, Professional, Developer Suite	Negates an expression.
Exponentiation (^)	View, Standard, Professional, Developer Suite	Raises an expression by a power.
Subtraction (-)	View, Standard, Professional, Developer Suite	Subtracts two expressions from each other.
Polarity Sign (#)	View, Standard, Professional, Developer Suite	Determines a code number for the polarity sign of an expression.

**Bitwise Operations**

Operator	Available in	Used for
Bitwise Boolean Exclusive OR	View, Standard, Professional, Developer Suite	Performs a bitwise Boolean Exclusive OR operation for two expressions.
Bitwise Boolean NOT (~)	View, Standard, Professional, Developer Suite	Determines the bitwise complement.
Bitwise Boolean OR ( )	View, Standard, Professional, Developer Suite	Performs a bitwise Boolean OR operation for two expressions.
Bitwise Boolean AND (&)	View, Standard, Professional, Developer Suite	Performs a bitwise Boolean AND operation for two expressions.
Bitwise Shift Right (>>)	View, Standard, Professional, Developer Suite	Shifts the bits in an integer to the right by the specified number of positions.
Bitwise Shift Left (<<)	View, Standard, Professional, Developer Suite	Shifts the bits in an integer to the left by the specified number of positions.

**Data Organization**

Operator	Available in	Used for
Bundle	View, Standard, Professional, Developer Suite	Bundles scalar values into data series, data series into data matrices or signals into signal series.
data series	View, Standard, Professional, Developer Suite	Forms a data series with linear ascending or descending values.
List	View, Standard, Professional, Developer Suite	Combines any values into a list.
Concatenation (:)	View, Standard, Professional, Developer Suite	Appends strings, scalar values, data series, data matrices, signals, signal series and space curves to one another.
Multiply (#)	View, Standard, Professional, Developer Suite	Multiplies a value by a given number.

**Data Conversion**

<b>Operator</b>	<b>Available in</b>	<b>Used for</b>
AbsoluteTime	View, Standard, Professional, Developer Suite	Converts time data to absolute time.
Boolean	View, Standard, Professional, Developer Suite	Converts an expression into the Boolean data type.
CalendarTime	View, Standard, Professional, Developer Suite	Converts an expression into the calendar time data type.
ComplexInteger16	View, Standard, Professional, Developer Suite	Converts an expression into the 16-bit complex integer data type.
ComplexInteger32	View, Standard, Professional, Developer Suite	Converts an expression into the 32-bit complex integer data type.
ComplexInteger64	View, Standard, Professional, Developer Suite	Converts an expression into the 64-bit complex integer data type.
ComplexFloatingPoint32	View, Standard, Professional, Developer Suite	Converts an expression into the 32-bit complex floating point value data type.
ComplexFloatingPoint64	View, Standard, Professional, Developer Suite	Converts an expression into the 64-bit complex floating point value data type.
FloatingPoint32	View, Standard, Professional, Developer Suite	Converts an expression into the 32-bit floating point value data type.
FloatingPoint64	View, Standard, Professional, Developer Suite	Converts an expression into the 64-bit floating point value data type.
Integer16	View, Standard, Professional, Developer Suite	Converts an expression into the 16-bit integer data type.
Integer32	View, Standard, Professional, Developer Suite	Converts an expression into the 32-bit integer data type.
Integer64	View, Standard, Professional,	Converts an expression into the 64-bit integer data type.

Operator	Available in	Used for
	Developer Suite	
RelativeTime	View, Standard, Professional, Developer Suite	Converts time data to relative time.
String	View, Standard, Professional, Developer Suite	Converts an expression into the string data type.
Unit	View, Standard, Professional, Developer Suite	Adds a unit to a value and converts it to a quantity or transforms the unit of a quantity.
Value	View, Standard, Professional, Developer Suite	Removes the unit from a quantity or evaluates an object reference.

Converts time data to absolute time.

### Syntax

**AbsoluteTime**[<Y|X|Z>] DataObject

or

**AbsoluteTimeStrict**[<Y|X|Z>] DataObject

The syntax of the **AbsoluteTime** operator consists of the following elements:

Element	Description
DataObject	The data object with data to be used and converted.

### Remarks

The conversion to absolute time is done by adding the timestamp of the data object on the **General** tab of the Properties dialog box for the data component that is to be converted.

The **AbsoluteTimeStrict** variant of the operator returns an error when compiling the formula if DataObject is an FScript expression that does not supply a data object. When executing the formula, this variant issues an error if the relevant data object component does not have a timestamp. The **AbsoluteTime** variant ignores both errors and does not change the argument.

The operator can process all data structures and real data types. If a component is appended to the operator, such as in **AbsoluteTime<X>**, only the relevant component is processed in aggregate data structures and the data of the other components remain unchanged in the result. If no component is appended, all components are processed. For non-aggregate data structures, **AbsoluteTime<Y>** is equivalent to **AbsoluteTime** and **AbsoluteTime<X>** as well as **AbsoluteTime<Z>** have no effect.

If the relevant data component is already in absolute time, i.e. it is of the calendar time data type, it will be copied without modification. If the operator is used for a particular component and the data object does not have a timestamp for the component to be converted, an error will be output. If Unit Manager is activated, it will also check whether the data component unit has time as the dimension. If it does not, no conversion will take place and it will be copied to the result unchanged.

Since the operator has to extract the timestamp from the data object, not just any expression is permitted as the argument. Only expressions that represent a data object are allowed. The data in the data object itself are not changed due to the operation but are just read out.

### Available in

FlexPro View, Standard, Professional, Developer Suite

### Examples

<b>AbsoluteTime&lt;X&gt;</b> Signal	Converts the X component of the signal to absolute time.
<b>AbsoluteTime</b> Signal	Converts all components of the signal with a time unit to absolute time.
<b>AbsoluteTime&lt;X&gt;</b> Signal(Amplitude, Time)	Generates an error message, since the operator argument does not represent a data object.

Converts time data to relative time.

**Syntax**

**RelativeTime**[<Y|X|Z>] Data Object

or

**RelativeTimeStrict**[<Y|X|Z>] Data Object

The syntax of the **RelativeTime**operator consists of the following elements:

Element	Description
Data Object	The data object with data to be used and converted.

**Remarks**

The conversion to relative time is done by subtracting the timestamp of the data object on the **General** tab of the Properties dialog box for the data component that is to be converted.

The **RelativeTimeStrict** variant of the operator returns an error when compiling the formula if Data Object is an FPScript expression that does not supply a data object. When executing the formula, this variant issues an error if the relevant data object component does not have a timestamp. The **RelativeTime** variant ignores both errors and does not change the argument.

The operator can process all data structures and real data types. If a component is appended to the operator, such as in **RelativeTime<X>**, only the relevant component is processed in aggregate data structures and the data of the other components remain unchanged in the result. If no component is appended, all components are processed. For non-aggregate data structures, **RelativeTime<Y>** is equivalent to **RelativeTime** and **RelativeTime<X>** as well as **RelativeTime<Z>** have no effect.

If the relevant data component is already in relative time, i.e. it is not of the calendar time data type, it will be copied without modification. If the data object does not have a timestamp for the component to be converted, the first value in the data set will be subtracted from all the data instead, i.e. the first value will be assigned the relative time of t = 0. If the operator is used for a particular component, the data object does not have a timestamp for the component to be converted and the first value in the data set is also void, an error will be output. If Unit Manager is activated, it will check whether the data component unit has time as the dimension. If it does not, no conversion will take place and it will be copied to the result unchanged.

Since the operator has to extract the timestamp from the data object, not just any



expression is permitted as the argument. Only expressions that represent a data object are allowed. The data in the data object itself are not changed due to the operation but are just read out.

### Available in

FlexPro View, Standard, Professional, Developer Suite

### Examples

<code>RelativeTime&lt;X&gt; Signal</code>	Converts the X component of the signal to relative time.
<code>RelativeTime Signal</code>	Converts all components of the signal with a time unit to relative time.
<code>RelativeTime&lt;X&gt; Signal(Amplitude, Time)</code>	Generates an error message, since the operator argument does not represent a data object.

### Complex Numbers

Operator	Available in	Used for
Imag	View, Standard, Professional, Developer Suite	Forms the imaginary component of a complex number.
Complex Number	View, Standard, Professional, Developer Suite	Combines a real component and an imaginary component into a complex number.
Conjugate Complex	View, Standard, Professional, Developer Suite	Forms the conjugate-complex expression.
Real	View, Standard, Professional, Developer Suite	Forms the real component of a complex number.

**Logic**

Operator	Available in	Used for
Logical NOT	Standard, Professional, Developer Suite	Carries out a logical Boolean NOT operation for an expression.
Logical OR	Standard, Professional, Developer Suite	Carries out a logical Boolean OR operation for two expressions.
Logical AND	Standard, Professional, Developer Suite	Carries out a logical Boolean AND operation for two expressions.

**Data Access**

Operator	Available in	Used for
Property	View, Standard, Professional, Developer Suite	Provides access to the properties of an object.
Index	View, Standard, Professional, Developer Suite	Extracts an individual value or a section from an expression.
Indirection	View, Standard, Professional, Developer Suite	Serves to access data sets, functions and variables by using their names.
Component	View, Standard, Professional, Developer Suite	Extracts the X, Y or Z component from a data set with an aggregate data structure.
List Element	View, Standard, Professional, Developer Suite	Takes an element from a list or selects a list element to which a new value is to be assigned. The operator can also be used to access a component of an aggregate data structure.
Value Index	View, Standard, Professional, Developer Suite	Extracts an individual value or a segment from a signal or a signal series. The indexing does not happen via indices, but instead via X or Z values.

**Comparison**

Operator	Available in	Used for
Equal To (==)	Standard,	Compares two numerical expressions and returns

Operator	Available in	Used for
	Professional, Developer Suite	TRUE if the left expression is equal to the right expression.
Greater Than Or Equal To ( $\geq$ )	Standard, Professional, Developer Suite	Compares two numerical expressions and returns TRUE if the left expression is greater than or equal to the right expression.
Greater Than ( $>$ )	Standard, Professional, Developer Suite	Compares two numerical expressions and returns TRUE if the left expression is greater than the right expression.
Less Than Or Equal To ( $\leq$ )	Standard, Professional, Developer Suite	Compares two numerical expressions and returns TRUE if the left expression is less than or equal to the right expression.
Less Than ( $<$ )	Standard, Professional, Developer Suite	Compares two numerical expressions and returns TRUE if the left expression is less than the right expression.
Not Equal To ( $\neq$ )	Standard, Professional, Developer Suite	Compares two numerical expressions and returns TRUE if the left expression is not equal to the right expression.

## FPScript Statements

### Conditional Execution

Statement	Available in	Used for
If...Then...Else	View, Standard, Professional, Developer Suite	Evaluates an expression and executes a series of statements, depending on the result.

### Loops

Statement	Available in	Used for
Do...While	View, Standard, Professional, Developer Suite	Executes a series of statements until a given condition is FALSE.

Statement	Available in	Used for
For Each Column...End	View, Standard, Professional, Developer Suite	Repeats a series of statements for all data series, i.e. columns, in a data matrix or all signals in a signal series with the option of concurrent execution.
For Each Row...End	View, Standard, Professional, Developer Suite	Repeats a series of statements for all rows of a data set.
For Each Value...End	View, Standard, Professional, Developer Suite	Repeats a series of statements, traversing all values of a data set.
For Each Element...End	View, Standard, Professional, Developer Suite	Repeats a series of statements for all items in a list with the option of concurrent execution.
For...End	View, Standard, Professional, Developer Suite	Repeats a series of statements several times until a loop counter reaches a certain value.
While...Do...End	View, Standard, Professional, Developer Suite	Executes a series of statements while a given condition is TRUE.

## Assignment

Statement	Available in	Used for
Append	View, Standard, Professional, Developer Suite	Appends additional data to a value stored in a variable.
Assignment	View, Standard, Professional, Developer Suite	Assigns a value to an object property, a variable or a component, or a list element therein.
Indexed Assignment	View, Standard, Professional, Developer Suite	Assigns a new value to a section in a data series or a data matrix.
Set	View, Standard, Professional, Developer Suite	Assigns an object reference to a variable or a list element.

**Other**

<b>Statement</b>	<b>Available in</b>	<b>Used for</b>
Arguments	View, Standard, Professional, Developer Suite	Declares arguments for a formula.
Dim	View, Standard, Professional, Developer Suite	Declares local variable for a formula.
Recalculate	View, Standard, Professional, Developer Suite	Forces recalculation of the FPScript formula the next time the result is requested.
Return	View, Standard, Professional, Developer Suite	Ends the calculation of a formula and passes the specified expression as a result.
Throw	View, Standard, Professional, Developer Suite	Throws an exception and returns the specified expression as the value of the exception.
Try...Catch...End	View, Standard, Professional, Developer Suite	Catches exceptions that occur in the statements included and enables their processing.
With Environment...Do...End	View, Standard, Professional, Developer Suite	Sets one or more FPScript environmental variables to a particular value and compiles a series of statements within the context of these set environmental variables or runs these within this context.

**FPScript Keywords**

<b>Keyword</b>	<b>Available in</b>	<b>Used for</b>
ActivatedFolder	View, Standard, Professional, Developer Suite	Provides access to the activated subfolder within a folder in the FlexPro project database.
Application	View, Standard, Professional, Developer Suite	Provides access to the "Application" automation object, which represents the FlexPro application.
As Object	View, Standard, Professional, Developer Suite	Interprets a path name as an object reference instead of a value.
This	View, Standard, Professional,	Provides access to the formula where the FPScript code is located.

Keyword	Available in	Used for
	Developer Suite	
ThisFPObj	View, Standard, Professional, Developer Suite	Provides access to the FlexPro object that executes an embedded FPScript code.
ThisObj	View, Standard, Professional, Developer Suite	Provides access to the automation object that executes an embedded FPScript code.

## As Object

Interprets a path name as an object reference instead of a value.

## Syntax

PathName **As Object**

The syntax of the **As Object** statement consists of the following elements:

Element	Description
PathName	The path name of a data set or other object for which an object reference is to be made.

## Available in

FlexPro View, Standard, Professional, Developer Suite

## Example

```
Dim Obj = DataSet As Object    assigns to the Obj variable a reference to DataSet.  
Obj.Unit                        The Unit property of the data set is then accessed.  
  
MyFunction(Formula As Object) passes an object reference to Formula to an FPScript  
                                function.
```

## FPScript Properties

Property	Available in	Used for
AssignedX	Standard, Professional, Developer Suite	Read access to the name of the data object assigned as the X component.
AssignedZ	Standard, Professional, Developer Suite	Read access to the name of the data object assigned as the Z component.
Author	Standard, Professional, Developer Suite	Read access to the name of a data object's author.
Comments	Standard, Professional, Developer Suite	Read access to the object's comments.
CommentsX	Standard, Professional, Developer Suite	Read access to the comments on a data object's X component. Read/write access to the comments on a current formula's X component.
CommentsY	Standard, Professional, Developer Suite	Read access to the comments on the data object's Y component. Read/write access to the comments on the current formula's Y component.
CommentsZ	Standard, Professional, Developer Suite	Read access to the comments on the data object's Z component. Read/write access to the comments on the current formula's Z component.
CreationTime	Standard, Professional, Developer Suite	Read access to an object's creation time.
Cursor.NameX	Standard, Professional, Developer Suite	Read access to a labeling of the X component of the curve under a bound cursor
Cursor.NameY	Standard, Professional, Developer Suite	Read access to a labeling of the Y component of the curve under a bound cursor
Cursor.NameZ	Standard, Professional, Developer Suite	Read access to a labeling of the Z component of the curve under a bound cursor
Cursor. PositionIndex	Standard, Professional, Developer Suite	Read access to the row index of a bound cursor.
Cursor.PositionX	Standard, Professional,	Read access to the X value of a cursor.

Property	Available in	Used for
	Developer Suite	
Cursor.PositionY	Standard, Professional, Developer Suite	Read access to the Y value of a cursor.
Cursor.PositionZ	Standard, Professional, Developer Suite	Read access to the Z value of a cursor.
Cursor. PositionZIndex	Standard, Professional, Developer Suite	Read access to the column index of a bound cursor.
Cursor.UnitX	Standard, Professional, Developer Suite	Read access to the unit of the X component of the curve under a bound cursor.
Cursor.UnitY	Standard, Professional, Developer Suite	Read access to the unit of the Y component of the curve under a bound cursor.
Cursor.UnitZ	Standard, Professional, Developer Suite	Read access to the unit of the Z component of the curve under a bound cursor.
FilePath	Standard, Professional, Developer Suite	Read access to the name and the path to the media file of a media.
Formula	Standard, Professional, Developer Suite	Read access to the FPScript code of a formula
FullName	Standard, Professional, Developer Suite	Read access to the name and the path in the project database of an object.
Markers. PositionCurve1	Standard, Professional, Developer Suite	Read access to the first curve numbers of the bound markers.
Markers. PositionCurve2	Standard, Professional, Developer Suite	Read access to the second curve numbers of the bound range markers.
Markers. PositionIndex1	Standard, Professional, Developer Suite	Read access to the first row indices of the bound markers.
Markers. PositionIndex2	Standard, Professional,	Read access to the second row indices of the bound range markers.



Property	Available in	Used for
	Developer Suite	
Markers.PositionX	Standard, Professional, Developer Suite	Read access to the X values of the markers.
Markers.PositionY	Standard, Professional, Developer Suite	Read access to the Y values of the markers.
Markers.PositionZ	Standard, Professional, Developer Suite	Read access to the Z values of the markers.
Markers. PositionZIndex1	Standard, Professional, Developer Suite	Read access to the first column indices of the bound markers.
Markers. PositionZIndex2	Standard, Professional, Developer Suite	Read access to the second column indices of the bound range markers.
ModificationTime	Standard, Professional, Developer Suite	Read access to the modification time of an object.
Name	Standard, Professional, Developer Suite	Read access to the name of an object. Read/write access to the name of the embedded FPScript.
Objects.Names	Standard, Professional, Developer Suite	Read access to the names of the objects in a folder.
Origin	Standard, Professional, Developer Suite	Read access to the origin of a data object.
Parameters	Standard, Professional, Developer Suite	Read access to a parameter of an object.
Parameters.Unit	Standard, Professional, Developer Suite	Read access to the unit of an object parameter.
Path	Standard, Professional, Developer Suite	Read access to the path in the project database of an object.
QuantityX	Standard, Professional,	Read access to the name of the physical quantity of a data object's X component. Read/write access to the

Property	Available in	Used for
	Developer Suite	name of the physical quantity of the current formula's X component.
QuantityY	Standard, Professional, Developer Suite	Read access to the name of the physical quantity of a data object's Y component. Read/write access to the name of the physical quantity of the current formula's Y component.
QuantityZ	Standard, Professional, Developer Suite	Read access to the name of the physical quantity of a data object's Z component. Read/write access to the name of the physical quantity of the current formula's Z component.
UnitX	Standard, Professional, Developer Suite	Read access to the unit of a data object's X component. Read/write access to the unit of the current formula's X component.
UnitY	Standard, Professional, Developer Suite	Read access to the unit of a data object's Y component. Read/write access to the unit of the current formula's Y component.
UnitZ	Standard, Professional, Developer Suite	Read access to the unit of a data object's Z component. Read/write access to the unit of the current formula's Z component.
TimestampX	Standard, Professional, Developer Suite	Read access to the unit of a data object's Z timestamp. Read/write access to the unit of the current formula's Z component.

### TimestampY

Read access to the timestamp on the data object's Y component. Read/write access to the timestamp on the current formula's Y component.

### Syntax

[DataObject].**TimestampY**

or

**.TimestampY** = Time

The syntax of the **TimestampY** property consists of the following elements:

Element	Description
DataObject	Optional. The data object being accessed. If you omit this element, then the current formula is accessed.
Time	A calendar time value to be assigned as the timestamp for the Y

Element	Description
	component of the current formula.

### Remarks

The timestamp is only relevant for time data. It provides the date and time assigned to the time  $t = 0$ . FlexPro uses this attribute to display the Y values of a data set in either relative time or absolute time in a diagram, for instance.

Write access to the header information of a data object is only allowed for the formula where the FPScript code is located. You should only use this option in embedded FPScript. For standard formula objects, you should edit the attribute on the **General** tab of the Properties dialog box instead.

### Available in

FlexPro Standard, Professional, Developer Suite

### Examples

```
\DataSet.TimestampY
```

Returns a calendar time value with the timestamp on a data set's Y component in the root folder.

```
.TimestampY = '1.1.2013 12:24'
```

Sets the the timestamp of the current formula's Y component to 1.1.2013 12:24.

### TimestampZ

Read access to the timestamp on the data object's Z component. Read/write access to the timestamp on the current formula's Z component.

### Syntax

```
[DataObject].TimestampZ
```

or

```
.TimestampZ = Time
```

The syntax of the **TimestampZ** property consists of the following elements:

Element	Description
DataObject	Optional. The data object being accessed. If you omit this element, then the current formula is accessed.
Time	A calendar time value to be assigned as the timestamp for the Z component of the current formula.

### Remarks

The timestamp is only relevant for time data. It provides the date and time assigned to the time  $t = 0$ . FlexPro uses this attribute to display the Z values of a data set in either relative time or absolute time in a diagram, for instance.

Write access to the header information of a data object is only allowed for the formula where the FPScript code is located. You should only use this option in embedded FPScript. For standard formula objects, you should edit the attribute on the **General** tab of the Properties dialog box instead.

### Available in

FlexPro Standard, Professional, Developer Suite

### Examples

`\DataSet.TimestampZ` Returns a calendar time value with the timestamp on a data set's Z component in the root folder.

`.TimestampZ = '1.1.2013 12:24'` Sets the the timestamp of the current formula's Z component to 1.1.2013 12:24.

## FPScript Functions

### Acoustics

Function	Available in	Used for
AcousticCalibration	Acoustics Option	Calculates the calibration value from a calibration signal.
FrequencyWeightingA	Acoustics Option	Calculates the frequency weighting A according to IEC 651 for an input signal or a signal series.

Function	Available in	Used for
FrequencyWeightingB	Acoustics Option	Calculates the frequency weighting B according to IEC 651 for an input signal or a signal series.
FrequencyWeightingC	Acoustics Option	Calculates the frequency weighting C according to IEC 651 for an input signal or a signal series.
Loudness	Acoustics Option	Calculates the loudness from third octave spectra (ISO 532:B) or octave spectra (ISO 532:A).
SoundLevel	Acoustics Option	Calculates sound levels from an input signal or an input signal series.
SoundPower	Acoustics Option	Calculates the sound power level. The level signals of several microphones are averaged energetically and a number of correction terms for barometric pressure, temperature, background noise, environmental correction and size of the measuring surface are considered.
TimeDomainOctaveAnalysis	Acoustics Option	Calculates an octave analysis of the input signal using time domain filters. The bandwidths octave, third octave, 1/6 octave, 1/12 octave and 1/24 octave can be selected. The frequency range within which the octave analysis is calculated can be selected within the limits of 1 Hz to 100 kHz.

### Importing Files

Function	Available in	Used for
ReadAstroMedDCRFile	View, Standard, Professional, Developer Suite	Reads a data set from a file in Astro-Med Dash 18/8X Data Capture Record format.
ReadBinaryFile	View, Standard, Professional, Developer Suite	Reads a data series from a binary file.
ReadBSME3File	View, Standard, Professional, Developer Suite	Reads a data set from a file in B+S ME 3 format V 3.00.
ReadCDFFile	View, Standard, Professional, Developer Suite	Reads a Z or R variable from a file in CDF format.
ReadDASYTECFile	View, Standard, Professional, Developer Suite	Reads a data set from a file in DASYTEC format from DASyLab.

Function	Available in	Used for
ReadDataFile	View, Standard, Professional, Developer Suite	Reads an element, such as a channel, from a data file.
ReadDPOFile	View, Standard, Professional, Developer Suite	Reads a signal from a file in Tektronix DPO 3D image file format.
ReadEDASFileX	View, Standard, Professional, Developer Suite	Reads the X component of a channel from a file in EDAS format.
ReadEDASFileY	View, Standard, Professional, Developer Suite	Reads the Y component of a channel from a file in EDAS format.
ReadFlexProFile	View, Standard, Professional, Developer Suite	Reads a data set from a FlexPro text file.
ReadGouldScopeFile	View, Standard, Professional, Developer Suite	Reads a data set from a file in Gould Oscilloscope format.
ReadINSIGHTFile	View, Standard, Professional, Developer Suite	Reads a data set from a file in INSIGHT's SDB or MDF format.
ReadIOtechFile	View, Standard, Professional, Developer Suite	Reads a data set from a file in IOtech's WaveView, DaqView, PersonalDaqView or ChartView format.
ReadLeCroyWaveformFile	View, Standard, Professional, Developer Suite	Reads a LeCroy Waveform file as a signal.
ReadMATLABFile	View, Standard, Professional, Developer Suite	Reads a field from a file in MATLAB format.
ReadNextViewFile	View, Standard, Professional, Developer Suite	Reads an analog or digital channel from a file in BMC NextView format.
ReadODBC	View, Standard, Professional, Developer Suite	Reads a column from a table or data source using ODBC.
ReadORFile	View, Standard, Professional, Developer Suite	Reads a channel from a file in Yokogawa ORP/ORM format.

Function	Available in	Used for
ReadOROSWaveFile	View, Standard, Professional, Developer Suite	Reads a channel from an OROS Wave file.
ReadPCScopeFileX	View, Standard, Professional, Developer Suite	Reads the X component of a channel from a file in IMTEC's PC-Scope format.
ReadPCScopeFileY	View, Standard, Professional, Developer Suite	Reads the Y component of a channel from a file in IMTEC's PCScope format.
ReadSEFRAMRecorderFile	View, Standard, Professional, Developer Suite	Reads an analog or digital channel from a file in SEFRAM recording format.
ReadSignalysFileX	View, Standard, Professional, Developer Suite	Reads the X component of a channel from a file in Signalys format.
ReadSignalysFileY	View, Standard, Professional, Developer Suite	Reads the Y component of a channel from a file in Signalys format.
ReadSPSSFile	View, Standard, Professional, Developer Suite	Reads a variable from an SPSS file.
ReadSYSTATFile	View, Standard, Professional, Developer Suite	Reads a variable from a SYSTAT file.
ReadTAFffmatFile	View, Standard, Professional, Developer Suite	Reads a data set from a file in TEAC TAFffmat format.
ReadTDSFile	View, Standard, Professional, Developer Suite	Reads a signal from a file in Tektronix TDS Waveform format.
ReadTEAMFile	View, Standard, Professional, Developer Suite	Reads a data set from a file in Nicolet Odyssey TEAM format.
ReadTextFile	View, Standard, Professional, Developer Suite	Reads a data set from a text file.
ReadTurboLabFileX	View, Standard, Professional, Developer Suite	Reads the X component of a channel from a file in TurboLab format.

Function	Available in	Used for
ReadTurboLabFileY	View, Standard, Professional, Developer Suite	Reads the Y component of a channel from a file in TurboLab format.
ReadWaveFile	View, Standard, Professional, Developer Suite	Reads an audio file as a signal or signal series.
ReadWTFFile	View, Standard, Professional, Developer Suite	Reads a data set from a file in Nicolet Waveform format.

## Data Information

Function	Available in	Used for
DataOrder	View, Standard, Professional, Developer Suite	Determines the data structure of a data set or of its X component.
DataSetStructure	Standard, Professional, Developer Suite	Determines the data structure of a data set.
DataSetType	Standard, Professional, Developer Suite	Determines the data type of a data set.
HasVoidValues	View, Standard, Professional, Developer Suite	Returns TRUE if a data set contains void values.
Increment	View, Standard, Professional, Developer Suite	Determines the increment of a data set or of its X component with linearly increasing or decreasing values.
ListElementName	View, Standard, Professional, Developer Suite	Determines the name of the list element with the specified index.
NumberOfColumns	View, Standard, Professional, Developer Suite	Determines the number of columns in a data set.
NumberOfElements	View, Standard, Professional, Developer Suite	Determines the number of elements in a data set.
NumberOfRows	View, Standard, Professional,	Determines the number of rows in a data set.



Function	Available in	Used for
Rank	Developer Suite	
	Standard, Professional, Developer Suite	Determines the number of dimensions of a data set.
SamplingRate	View, Standard, Professional, Developer Suite	Determines the sampling rate of a data set or of its X component with linearly increasing or decreasing values.
Shape	Standard, Professional, Developer Suite	Determines the dimension lengths of a data set.
SIUnits	Standard, Professional, Developer Suite	Returns the presentation of the unit of a data set in SI units.
Trend	Standard, Professional, Developer Suite	Determines the constant, linear or adaptive trend of a data set.
Unit	Standard, Professional, Developer Suite	Returns the unit of a data set as a quantity with the value 1.
UnitSymbol	Standard, Professional, Developer Suite	Returns the symbol of the data set unit.
UnitType	Standard, Professional, Developer Suite	Determines the type of unit for a data set.

## Data Manipulation

Function	Available in	Used for
Absolute	Standard, Professional, Developer Suite	Forms the absolute value of real numbers, complex numbers or time spans.
AdjustUnit	Standard, Professional, Developer Suite	Adjusts the unit(s) of a data set to those of a different data set.
AssignHeader	View, Standard, Professional, Developer Suite	Assigns header information to a value.
Bit	View, Standard,	Extracts a bit trace from a data set with digital data.

Function	Available in	Used for
ChangeDataType	Professional, Developer Suite	
ChangeUnit	Standard, Professional, Developer Suite	Converts the data type of a data set.
ChangeUnitSymbol	Standard, Professional, Developer Suite	Converts or sets the unit(s) of a data set.
Clip	Standard, Professional, Developer Suite	Replaces one or more unit symbols of a data set.
Clip2D	Standard, Professional, Developer Suite	Clips the amplitude of a data set.
CurveToSurface	Standard, Professional, Developer Suite	Cuts surfaces at an edge or along a closed curve.
Detrend	Standard, Professional, Developer Suite	Cuts surfaces at an edge or along a closed curve.
ImaginaryPart	Standard, Professional, Developer Suite	Transforms a space curve into a surface (signal series with Z component).
List	Standard, Professional, Developer Suite	Removes a constant, linear or adaptive trend from a data set.
ListToSeries	View, Standard, Professional, Developer Suite	Forms the imaginary part of real or complex numbers.
NextHighestInteger	Standard, Professional, Developer Suite	Generates a list.
NextLowestInteger	Standard, Professional, Developer Suite	Converts a list with items of the same structure into a data series, data matrix or signal series.
Phase	Standard,	Determines the next highest integer for a floating point value.
		Determines the next lowest integer for a floating point value.
		Forms the phase of complex numbers.

Function	Available in	Used for
PrimaryListElement	Professional, Developer Suite	
RealPart	View, Standard, Professional, Developer Suite	Takes the primary element from a list.
RemoveHeader	Standard, Professional, Developer Suite	Forms the real part of real or complex numbers.
Reshape	View, Standard, Professional, Developer Suite	Removes header information from a value.
Round	View, Standard, Professional, Developer Suite	Changes the dimension lengths and/or number of dimensions of a data series or a data matrix.
RoundDown	Standard, Professional, Developer Suite	Rounds to a number of decimal places that you can specify.
RoundUp	Standard, Professional, Developer Suite	Rounds down to a number of decimal places that you can specify.
Sort	Standard, Professional, Developer Suite	Rounds up to a number of decimal places that you can specify.
SurfaceToCurve	Standard, Professional, Developer Suite	Sorts a data set or provides a sort index to a data set.
XScale	Standard, Professional, Developer Suite	Transforms a surface (signal series with Z component) into a space curve.
	View, Standard, Professional, Developer Suite	Scales the X component of a signal or signal series.

### Dialog Boxes

Function	Available in	Used for
ChooseFile	Standard, Professional, Developer Suite	Opens a dialog box for choosing a file.

Function	Available in	Used for
Input	Standard, Professional, Developer Suite	Opens a dialog box for entering an expression.
MessageBox	Standard, Professional, Developer Suite	Displays a message box.
TextInput	Standard, Professional, Developer Suite	Opens a dialog box for entering text.
TimeInput	Standard, Professional, Developer Suite	Opens a dialog box for entering the date and time.
TimeSpanInput	Standard, Professional, Developer Suite	Opens a dialog box for entering a time span.

## Event Isolation

Function	Available in	Used for
Bursts	Standard, Professional, Developer Suite	Searches for beginnings and/or ends of bursts in a data set.
Extrema	Standard, Professional, Developer Suite	Searches for local minima and/or maxima in a data set.
GlobalExtrema	Standard, Professional, Developer Suite	Searches for global minima and/or maxima in a data set.
GlobalMaximum	Standard, Professional, Developer Suite	Searches for the global maximum in a data set.
GlobalMinimum	Standard, Professional, Developer Suite	Searches for the global minimum in a data set.
IndexAfter	Standard, Professional, Developer Suite	Performs a sequential operation for two index data sets.
IndexAnd	Standard, Professional,	Performs a Boolean AND operation for two index data sets.

Function	Available in	Used for
IndexBefore	Developer Suite	
IndexNot	Standard, Professional, Developer Suite	Performs a sequential operation for two index data sets.
IndexOr	Standard, Professional, Developer Suite	Forms the complement of an index data set.
Intersections	Standard, Professional, Developer Suite	Performs a logical Boolean OR operation for two index data sets.
LevelCrossings	Standard, Professional, Developer Suite	Determines the exact intersections of two data sets or level crossings of a data set.
NegativePeaks	Standard, Professional, Developer Suite	Searches a data set for level crossings.
NextBurst	Standard, Professional, Developer Suite	Searches in a data set for negative peaks.
NextExtremum	Standard, Professional, Developer Suite	Searches for the next beginning and/or end of a burst from a given position onwards.
NextGlobalExtremum	Standard, Professional, Developer Suite	Searches for the next extreme from a given position onwards.
NextLevelCrossing	Standard, Professional, Developer Suite	Searches for the next global extreme from a given position onwards.
NextNegativePeak	Standard, Professional, Developer Suite	Searches for the next level crossing from a given position onwards.
NextPositivePeak	Standard, Professional, Developer Suite	Searches in a data set for the next negative peak starting from a specified position.
NextSlope	Standard, Professional,	Searches in a data set for the next positive peak starting from a specified position.
		Searches for the next slope of a specified minimum steepness from a certain position onwards.

Function	Available in	Used for
	Developer Suite	
NextSlopeAtLevel	Standard, Professional, Developer Suite	Searches for the next slope of a specified minimum steepness through a level from a certain position onwards.
NextValueAboveLevel	Standard, Professional, Developer Suite	Searches for the next value above a certain level.
NextValueBelowLevel	Standard, Professional, Developer Suite	Searches for the next value below a certain level.
NextValueInBurst	Standard, Professional, Developer Suite	Searches for the next value in a burst starting from a given position.
NextValueInInterval	Standard, Professional, Developer Suite	Searches for the next value in a particular interval or for the next void value.
PositivePeaks	Standard, Professional, Developer Suite	Searches in a data set for positive peaks.
SearchValue	Standard, Professional, Developer Suite	Searches for one or more values in a data set.
Slopes	Standard, Professional, Developer Suite	Searches for slopes of a specified minimum steepness in a data set.
SlopesAtLevel	Standard, Professional, Developer Suite	Searches within a data set for slopes of a specified minimum steepness that have to pass through a particular level.
Trigger	Standard, Professional, Developer Suite	Represents a Schmitt trigger comparator, which performs a threshold observation for a data set.
ValuesAboveLevel	Standard, Professional, Developer Suite	Searches a data set for values above a certain level.
ValuesBelowLevel	Standard, Professional, Developer Suite	Searches a data set for values below a certain level.
ValuesInBursts	Standard, Professional,	Searches within a data set for values that belong to a burst.

Function	Available in	Used for
ValuesInInterval	Developer Suite View, Standard, Professional, Developer Suite	Searches a data set for values that lie in a particular interval, or searches for void values.

## Filtering and Smoothing

Function	Available in	Used for
AmplitudeResponse	Digital Filters Option	Calculates the amplitude response from the filter coefficients.
CFCFilter	Digital Filters Option	Filters a data set using a CFC filter. CFC is short for Channel Frequency Class.
Filters	Standard, Professional, Developer Suite	Filters a data set with a Finite Impulse Response (FIR) or an Infinite Impulse Response (IIR) filter.
FIRFilterEquiripple	Digital Filters Option	Calculates the impulse response of an FIR band filter using the equiripple algorithm.
FIRFilterWindow	Digital Filters Option	Calculates an FIR filter's impulse response using windowing.
GroupDelay	Digital Filters Option	Calculates the group delay from the filter coefficients.
IIRFilter	Standard, Professional, Developer Suite	Calculates the numerator and denominator coefficients or the poles and zeros of an infinite impulse response filter (IIR filter).
LoessFilter	Digital Filters Option	Filters a data set with a Loess smoothing filter (locally weighted regression scatter plot smoothing).
LowessFilter	Digital Filters Option	Filters a data set with a Lowess smoothing filter (locally weighted regression scatter plot smoothing).
PhaseResponse	Digital Filters Option	Calculates the phase response from the filter coefficients.
SavitzkyGolayFilter	Digital Filters Option	Filters a data set using a Savitzky-Golay FIR smoothing filter.
Smooth	Standard, Professional, Developer Suite	Smooths a data set by calculating a floating mean value.
VibrationFrequencyWeighting	Human Body	Filters an acceleration signal with a band-

Function	Available in	Used for
	Vibrations Option	limited weighting filter for analyzing whole-body and hand-transmitted vibrations.

## FPScript

Function	Available in	Used for
Conditional	Standard, Professional, Developer Suite	Data analysis. Functional variant of the If... Then...Else statement.
Execute	Standard, Professional, Developer Suite	Interprets a string as FPScript code and executes it.

## Envelope Curves

Function	Available in	Used for
LowerEnvelope	Standard, Professional, Developer Suite	Determines a lower envelope for a signal or the indices of points in a signal that belong to its lower envelope.
UpperEnvelope	Standard, Professional, Developer Suite	Determines an upper envelope for a signal or the indices of points in a signal that belong to its upper envelope curve.

## Curve Interpolation

Function	Available in	Used for
CompensatingSpline	Standard, Professional, Developer Suite	Interpolates a data set through a compensating spline curve and samples this at definable points.
ParametricSpline	Standard, Professional, Developer Suite	Interpolates a data set through a parametric spline curve and samples it at definable points.
PeriodicSpline	Standard, Professional, Developer Suite	Interpolates a data set using a periodic spline curve and samples the spline curve at definable points.
PolynomialInterpolation	Standard, Professional, Developer Suite	Carries out polynomial interpolation at predefinable sampling points. Polynomial interpolation is understood to be the search for a polynomial that



Function	Available in	Used for
Spline	Standard, Professional, Developer Suite	runs exactly through the predefined points.  Interpolates a data set with a spline curve and samples this curve at definable points.

## Surface Interpolation

Function	Available in	Used for
BicubicSpline	Standard, Professional, Developer Suite	Interpolates a two-dimensional data set through a bicubic spline surface and samples this spline surface at definable points.
CompensatingSpline2D	Standard, Professional, Developer Suite	Interpolates a two-dimensional data set through a grid of compensating spline curves and samples these curves at definable points.
GriddedSurface	Standard, Professional, Developer Suite	Interpolates a two-dimensional data set using Natural Neighbor interpolation and samples this at definable points.
Isoline	Standard, Professional, Developer Suite	Determines one or more isolines from a data matrix or signal series.
ScatteredSurface	Standard, Professional, Developer Suite	Models a surface using the Natural Neighbor method for the sampling points specified as a space curve and evaluates this at definable grid points.
Spline2D	Standard, Professional, Developer Suite	Interpolates a two-dimensional data set through a grid of spline curves and samples these curves at definable points.

## Calendar Time

Function	Available in	Used for
CurrentDate	View, Standard, Professional, Developer Suite	Determines the calendar time value of the current time with a precision of one second.
Date	View, Standard, Professional, Developer Suite	Converts a local time into a calendar time value where the date and time must be specified.
Day	View, Standard, Professional,	Determines the day of the month of one or more calendar time values or the number of days of one or

Function	Available in	Used for
DayOfYear	Developer Suite View, Standard, Professional, Developer Suite	more time span values. Determines the day in the year of one or more calendar time values.
Duration	View, Standard, Professional, Developer Suite	Generates a time span value using the elements provided.
Hour	View, Standard, Professional, Developer Suite	Determines the hour of the day of one or more calendar time or time span values.
Minute	View, Standard, Professional, Developer Suite	Determines the minute in the hour of one or more calendar time or time span values.
Month	View, Standard, Professional, Developer Suite	Determines the month in the year of one or more calendar time values.
Second	View, Standard, Professional, Developer Suite	Determines the second in the minute of one or more calendar time or time span values.
Time	View, Standard, Professional, Developer Suite	Converts a local time into a calendar time value. You only have to specify the time. Month, day and year are provided by the system time.
TimeFilter	Standard, Professional, Developer Suite	Searches a data set with calendar time values for periodic time segments.
WeekDay	View, Standard, Professional, Developer Suite	Determines the week day for one or more calendar time values.
Year	View, Standard, Professional, Developer Suite	Determines the value for the year of one or more calendar time values.

## Counting

Function	Available in	Used for
CompoundMaximumValueCount	Counting Procedures Option	Performs a maximum value count in accordance with the DIN 45667 standard synchronously for two signals.
CompoundSamplingCount	Counting	Carries out a sampling count in accordance with the

Function	Available in	Used for
nt	Procedures Option	DIN 45667 standard synchronously for two signals.
CompoundTimeAtLevel Count	Counting Procedures Option	Performs a time-at-level count in accordance with the DIN 45667 standard synchronously for two signals.
Histogram	Standard, Professional, Developer Suite	Creates a histogram for a class division that you can specify.
LevelCrossingCount	Counting Procedures Option	Carries out a level crossing count from the Markov or Rainflow Matrix. The function counts the absolute frequencies of positive and negative level crossings.
MarkovMatrix	Counting Procedures Option	Determines the Markov Matrix (transfer matrix) for a data set.
MaximumValueCount	Counting Procedures Option	Carries out a maximum value count in accordance with the DIN 45667 standard.
MeanValueCount	Counting Procedures Option	Counts a signal and determines the class mean values.
PeakCount	Counting Procedures Option	Performs a peak value count from the Rainflow Matrix or Markov Matrix. The function counts the absolute or cumulated frequencies of peak or trough values.
RainflowMatrix	Counting Procedures Option	Determines the Rainflow Matrix for a data set.
RangeCount	Counting Procedures Option	Performs a range count from the Markov Matrix or a range pair count from the Rainflow Matrix. The function counts the absolute or cumulated frequencies of positive or negative ranges or range pairs (hystereses).
SamplingCount	Counting Procedures Option	Performs a sampling count in accordance with the DIN 45667 standard.
TimeAtLevelCount	Counting Procedures Option	Performs a time-at-level count for a signal in accordance with the DIN 45667 standard.

## Curve Fitting

Function	Available in	Used for
Approximation	Standard, Professional, Developer Suite	Approximates a linear model $Y(X)$ to given data according to the method of the least squares.

Function	Available in	Used for
Approximation2D	Standard, Professional, Developer Suite	Approximates a linear model $Y(X, Z)$ with two independent variables to given 2D data according to the least squares method.
NonLinearCurveFit	Standard, Professional, Developer Suite	Approximates a non-linear model to a data set and passes the model parameters found, the model function, and the statistical results.
NonLinearModel	Standard, Professional, Developer Suite	Calculates a non-linear model function for model parameters that you specify.
ParameterEstimation	Standard, Professional, Developer Suite	Approximates a non-linear model to a data set. Here, non-iterative methods are used. The available methods are the "Random Search" methods.

Logarithm and Exponent

Function	Available in	Used for
Exp	Standard, Professional, Developer Suite	Raises e to the power of the specified exponent.
Log	Standard, Professional, Developer Suite	Calculates the logarithm to the base e (natural logarithm).
Log10	View, Standard, Professional, Developer Suite	Calculates the logarithm to base 10.

General Mathematics

Function	Available in	Used for
Factorial	Standard, Professional, Developer Suite	Calculates the factorial $N!$ of a natural number.
J0	Standard, Professional, Developer Suite	Calculates the Bessel function of the first type of the order 0.

Function	Available in	Used for
J1	Standard, Professional, Developer Suite	Calculates the Bessel function of the first type of the order 1.
Jn	Standard, Professional, Developer Suite	Calculates the Bessel function of the first type with the order specified.
Product	Standard, Professional, Developer Suite	Calculates the product of all of the values in a data series or the products of all rows in a data matrix.
Sinc	Standard, Professional, Developer Suite	Calculates the sinc function $\text{Sin}(\text{Angle})/\text{Angle}$ .
Sqrt	Standard, Professional, Developer Suite	Calculates the square root.
TrackDistance	View, Standard, Professional, Developer Suite	Calculates the distance between locations on Earth.
Y0	Standard, Professional, Developer Suite	Calculates the Bessel function of the second type of the order 0.
Y1	Standard, Professional, Developer Suite	Calculates the Bessel function of the second type of the order 1.
Yn	Standard, Professional, Developer Suite	Calculates the Bessel function of the second type with the order specified.

## Matrices

Function	Available in	Used for
InverseMatrix	Standard, Professional, Developer Suite	Inverts a matrix.
MatrixMultiplication	Standard, Professional, Developer Suite	Multiplies matrices with vectors and matrices in all combinations.
TransposeMatrix	Standard, Professional,	Transposes a matrix.

Function	Available in	Used for
	Developer Suite	

### Signal Sampling

Function	Available in	Used for
DeltaCompress	Standard, Professional, Developer Suite	Reduces the number of values in a data set through delta compression.
Expand	Standard, Professional, Developer Suite	Increases the number of values in a data set.
LinearInterpolation	Standard, Professional, Developer Suite	Carries out a linear interpolation of a data set.
Reduce	View, Standard, Professional, Developer Suite	Reduces the number of values in a data set.
Resample	Standard, Professional, Developer Suite	Increases or reduces a data set's sampling rate.
Sample	Standard, Professional, Developer Suite	Samples a signal using linear interpolation.

### Signal Analysis

Function	Available in	Used for
ACF	Standard, Professional, Developer Suite	Calculates the autocorrelation function of a signal.
CCF	Standard, Professional, Developer Suite	Calculates the cross-correlation function of two signals.
Convolution	Standard, Professional, Developer Suite	Calculates the convolution product of two signals.
CumulativeSum	Standard, Professional,	Calculates the cumulative sum of a data set.

Function	Available in	Used for
Derivative	Developer Suite View, Standard, Professional, Developer Suite	Calculates the first derivative of a signal.
Frequency	Standard, Professional, Developer Suite	Determines the frequency of a data set.
ImpulseToFrequency	Standard, Professional, Developer Suite	Converts an impulse data set into a frequency data set.
Integral	View, Standard, Professional, Developer Suite	Calculates the integral of a signal.
Period	Standard, Professional, Developer Suite	Determines the period of a data set.

### Signal Generation

Function	Available in	Used for
Chirp	Standard, Professional, Developer Suite	Calculates a swept-frequency cosine, i.e. a cosine signal with a variable frequency.
Dirichlet	Standard, Professional, Developer Suite	Calculates the Dirichlet function, i.e. the periodic sinc function.
Noise	Standard, Professional, Developer Suite	Generates equally, normally or exponentially distributed random numbers.
Sawtooth	Standard, Professional, Developer Suite	Calculates the sawtooth function with a period of $2\pi$ , amplitude of 1 and given width.
Series	Standard, Professional, Developer Suite	Forms a data series with linear ascending or descending values.
Signal	View, Standard, Professional, Developer Suite	Combines a signal, signal series or a space curve from individual components.

Function	Available in	Used for
Square	Standard, Professional, Developer Suite	Calculates the square function with a period of $2\pi$ , amplitude of 1 and given duty cycle.
StraightLine	Standard, Professional, Developer Suite	Sets up a straight line equation based on the Y intercept and slope or based on two XY value pairs and evaluates this at the specified X positions.

## Spectral Analysis

Function	Available in	Used for
ApplyWindow	Standard, Professional, Developer Suite	Applies a window to the argument.
ARMASpectrum	Spectral Analysis Option	Computes the Autoregressive-Moving Average (ARMA) high-resolution frequency estimation spectrum.
ARSpectrum	Spectral Analysis Option	Computes the Autoregressive or AR Spectral estimators spectrum.
CepstralAnalysis	Spectral Analysis Option	Computes the cepstrum or its minimum phase reconstruction.
Coherence	Spectral Analysis Option	Computes the coherence spectrum between two data sets.
CrossPeriodogram	Spectral Analysis Option	Computes the windowed Fourier cross-periodogram between two data sets.
CrossSpectrum	Spectral Analysis Option	Computes the windowed Fourier cross-spectrum between two data sets.
CWTSpectrum	Spectral Analysis Option	Computes the Continuous Wavelet Transform (CWT) time-frequency spectrum.
DataWindow	Spectral Analysis Option	Creates a uniformly spaced data taper of a given size.
EigenSpectrum	Spectral Analysis Option	Computes the EigenAnalysis Spectral estimators spectrum.
FFTN	Standard, Professional, Developer Suite	Computes the complex Fourier transform of the argument.
FourierSpectrum	Standard, Professional, Developer Suite	Computes the windowed Fourier spectrum.



Function	Available in	Used for
FourierSpectrumUneven	Spectral Analysis Option	Computes the windowed Fourier spectrum for unevenly sampled data.
HarmonicEstimation	Spectral Analysis Option	Estimates the least-squares harmonic components of a signal.
IFFTn	Standard, Professional, Developer Suite	Computes the complex inverse Fourier transform of the argument.
IRFFTn	Standard, Professional, Developer Suite	Computes the real inverse Fourier transform of the argument.
MultitaperSpectrum	Spectral Analysis Option	Computes the Slepian or DPSS Multitaper Fourier spectrum.
OctaveAnalysis	Standard, Professional, Developer Suite	Performs an octave analysis for an amplitude spectrum.
OrderTracking	Order Tracking Option	Calculates an order tracking analysis for speed-dependent oscillations. For tracking analysis, the oscillatory signals measured at a certain speed are subjected to a Fourier transform (FFT). Individual spectral lines whose frequencies correspond to a multiple of the base frequency determined by the speed are then taken from the spectrum.
Periodogram	Standard, Professional, Developer Suite	Computes the segmented-overlapped Fourier spectrum.
SDOFResponse	Spectral Analysis Option	Calculates the responses of individual single-degree-of-freedom (SDOF) systems of the SRS model from the signal of an acceleration sensor.
SRS	Spectral Analysis Option	Calculates one or more shock response spectra (SRS) from the signal of an acceleration sensor.
SRSFromSDOFResponse	Spectral Analysis Option	Calculates the shock response spectrum (SRS) from the responses of the SRS model's individual SDOF systems, which can be calculated using the SDOFResponse function.
STFTSpectrum	Standard, Professional, Developer Suite	Computes the Short-Time Fourier Transform (STFT) time-frequency spectrum.
ThirdOctaveAnalysis	Standard, Professional,	Carries out a third octave analysis for an amplitude spectrum.

Function	Available in	Used for
	Developer Suite	
TransferFunction	Spectral Analysis Option	Computes the transfer function between an input and an output data set.
VarWindow	Spectral Analysis Option	Creates a data taper compatible with a data set.

## Statistics

Function	Available in	Used for
AbsoluteDeviationFromMean	Standard, Professional, Developer Suite	Determines the mean absolute deviation from the mean value of a data set. The displacements from the mean are determined for all values of the data set and the average is calculated from these.
AbsoluteDeviationFromMedian	Standard, Professional	Determines the mean absolute deviation from the median of a data set. This is the average of the deviations from the median. The median is the value that lies exactly in the center of a data series after sorting. If the data set contains an even number of values, the average of the two middle values is formed.
ANOVA	Inductive Statistics Option	Carries out a Fisher analysis of variance. You can either calculate an ANOVA table or perform an F-test. The F-test specifies whether the variances of multiple samples are significantly different or not. The ANOVA table provides characteristic quantities for the analysis of variance.
BartlettTest	Inductive Statistics Option	Carries out a Bartlett variance test.
ChiSquareTest	Inductive Statistics Option	Carries out a Chi-square goodness-of-fit test. The test checks whether the sample passed corresponds to the normal or exponential distribution specified.
ConfidenceInterval	Inductive Statistics Option	Calculates a confidence interval for the expectation value or the variance of the normal distribution population to which the specified sample belongs.
Correlation	Inductive	Determines the correlation matrix of a data

Function	Available in	Used for
Covariance	Statistics Option Inductive Statistics Option	matrix or the correlation of two data sets. Determines the covariance matrix of a data matrix or the covariance between two data sets.
DavidHartleyPearsonTest	Inductive Statistics Option	Carries out a David-Hartley-Pearson outlier test. The test checks whether the highest or lowest value in the normally distributed sample passed belongs to the sample or not.
Distribution	Inductive Statistics Option	Calculates various distribution or density functions.
GrubbsBeckTest	Inductive Statistics Option	Carries out a Grubbs-Beck outlier test. The test checks whether the highest and/or lowest value in the sample belongs to the sample or not.
KolmogorovSmirnovTest	Inductive Statistics Option	Carries out a Kolmogorov-Smirnov goodness-of-fit test. The test checks whether the sample passed corresponds to the normal or exponential distribution specified.
Maximum	View, Standard, Professional, Developer Suite	Determines individual, floating or block maxima for a data set.
Mean	View, Standard, Professional, Developer Suite	Determines individual, floating or block mean values for a data set.
Minimum	View, Standard, Professional, Developer Suite	Determines individual, floating and block minima for a data set.
Quantiles	Standard, Professional, Developer Suite	Determines individual, floating and block quantiles for a data set.
Range	View, Standard, Professional, Developer Suite	Determines global, floating or block ranges, i. e. differences between maxima and minima for a data set.
Sum	View, Standard, Professional, Developer Suite	Calculates the sum or square sum of all values in a data set.
Variance	View, Standard, Professional, Developer Suite	Determines the variance $\sigma^2$ of a data series.

**Trigonometry**

Function	Available in	Used for
ArcCos	Standard, Professional, Developer Suite	Calculates the inverse cosine of the argument. The result is output in radians.
ArcSin	Standard, Professional, Developer Suite	Calculates the inverse sine of the argument. The result is output in radians.
ArcTan	Standard, Professional, Developer Suite	Calculates the inverse tangent of the argument.
Cos	Standard, Professional, Developer Suite	Calculates the cosine of an angle.
CosHyp	Standard, Professional, Developer Suite	Calculates the hyperbolic cosine of a number.
Sin	Standard, Professional, Developer Suite	Calculates the sine of an angle.
SinHyp	Standard, Professional, Developer Suite	Calculates the hyperbolic sine of a number.
Tan	Standard, Professional, Developer Suite	Calculates the tangent of an angle.
TanHyp	Standard, Professional, Developer Suite	Calculates the hyperbolic tangent of a number.

**Void Values**

Function	Available in	Used for
InterpolateVoidValues	Standard, Professional, Developer Suite	Interpolates or extrapolates void floating point values in a data set.
RemoveVoidValues	Standard, Professional, Developer Suite	Removes void values from a data set.
SearchVoidValues	Standard,	Searches for void floating point values in a data set

Function	Available in	Used for
	Professional, Developer Suite	and passes their positions as a data series with indices.

## Strings

Function	Available in	Used for
Format	Standard, Professional, Developer Suite	Formats a value and returns the result as a string.
StringFind	Standard, Professional, Developer Suite	Searches for a substring and returns its position.
StringLeft	Standard, Professional, Developer Suite	Takes a substring out of a string starting from the left.
StringLength	Standard, Professional, Developer Suite	Determines the number of characters in a string.
StringLowerCase	Standard, Professional, Developer Suite	Converts a string into lower case.
StringMid	Standard, Professional, Developer Suite	Takes a substring out of a string starting from a particular position.
StringReplace	Standard, Professional, Developer Suite	Searches for a substring and replaces all occurrences with another string.
StringRight	Standard, Professional, Developer Suite	Takes a substring out of a string starting from the right.
StringSet	Standard, Professional, Developer Suite	Replaces a substring from a specific position in a string onwards.
StringUpperCase	Standard, Professional, Developer Suite	Converts a string into upper case.

## 6 Presenting Data

To present data and calculated results, FlexPro offers you the 2D and 3D diagram, column and cell table and text presentation objects. You can either export these objects directly or insert them in documents and worksheets.

Presentation objects have a definable fixed size corresponding to a rectangular area that the object adopts when it is placed into a document, for instance. However, if you open a presentation object in its own window, then its size is automatically adapted to fit the size of the window when you enable the appropriate option.

### Presentation and Document Templates

You can create templates for diagrams, tables, texts and documents and store these in the current project database or in a template database. You can then choose a previously saved template in the various wizards for creating objects.

You can embed or link diagrams, tables and text in document templates. Linked objects are replaced by the selected objects when the template is used at a later date. The links in the template serve as placeholders for the diagrams, tables and text to be inserted. For embedded presentation objects, however, the data links are replaced. If a document template contains an embedded diagram with two curves, for instance, then the selected data objects are entered in the curves when the template is used.

FlexPro makes a distinction between static and dynamic table and diagram templates. If you use a dynamic template, then a curve or a table column or table cell is created for each selected data set. In a static template, only the existing curves, columns or cells are linked to the selected data objects. Excess data objects, if present, are ignored. You define the type of template when saving it.

### Linking and Embedding

You can embed presentation objects into documents and worksheets or insert a link to a project database object. An embedded object is a component of the document or worksheet and can only be used in the location where it has been inserted. Links to embedded objects cannot be created.

If you would like to use the same object in several places, such as in a document and in a worksheet, then you have to set it up in the project database and insert a link to it at the points where it is to appear. When selected, linked objects are displayed with a dashed border and embedded objects are displayed with a solid border.

If you edit an object with links to it, then the changes affect all links. Updating various links does not take place automatically, but instead must be activated using the Update command. The exception here is the size of the object. You can set the dimensions for

each link individually. A diagram can therefore be displayed, for example, in a different size in the worksheet than the size in the document.

Not only can you insert links into FlexPro documents and worksheets, but you can also insert them into documents you created using a different application. For example, you can insert a link to a diagram into a text document. Your word processing program just needs an OLE interface. You can also update and edit these links.

You cannot embed presentation objects in external applications, because the FlexPro objects can only be used practically within the environment of the data and analyses on which they are based. However, you can embed a static representation of the object. This then only consists of the image information and cannot be edited subsequently using FlexPro.

### **Drawing in Presentation Objects**

You can insert any number of drawings into all presentation objects except the text object in order to highlight or label interesting parts of a curve, for instance. You can use the same set of drawing objects as in a document. All drawings that you create within a presentation object appear in the foreground. If you enlarge a diagram, the drawing will automatically be enlarged as well.

### **Multi-page Diagrams and Column Tables**

FlexPro can split 2D diagrams and column tables over several pages in a document. You can use this option, for instance, to display longer continuous curves in a diagram or to display larger data sets in a column table.

In a 2D diagram, page splitting and numbering depends on the options selected on the **Page Splitting** tab of the Properties dialog box for the diagram. For a column table, the space available on the first page is used and then wraps automatically onto the additional pages if not all of the data can be displayed.

You can place the diagram or the column table on any page in the document. The additional pages automatically appear at the same position on the subsequent pages of the document. The number of pages in the document is automatically adjusted by FlexPro so that all pages of the diagram or column table can be displayed. If you would like to edit or position a multi-page object, you need to switch to the page in the document where the first part of the object is displayed.

---

**Note:** You can only display entire multi-page column tables and diagrams in FlexPro documents and in the object window. In the worksheet or external applications, only the first page is displayed.

---

## Embedded FPScript

On the **Data** tab of the Properties dialog box for presentation objects, specify the path name of the formula or data set providing the data to be displayed, for instance, as a curve in a diagram. The easiest thing to do is to specify only the name of the data set. However, if you prefer, you can enter any FPScript expression. This embedded FPScript is then interpreted as a formula located in the same folder where the presentation object is also located.

The following FPScript expression can be used, for instance, to display the last signal of a signal series as a curve:

```
SignalSeries[-1]
```

To work with the FPScript expression, FlexPro dynamically creates a formula embedded in the presentation object, which is then compiled and executed. This formula, as with standard formulas, has the properties UnitY, UnitX, UnitZ, CommentsY, CommentsX, and CommentsZ. FlexPro attempts to create relevant entries for these properties automatically by taking these from the last data object used in the formula. In the example above, this would be the `Signal Series` data set. For the most frequent situation where the embedded FPScript only accesses a data set, this provides the correct entries, such as for axis labeling. If this is not the case, you will have to correct the relevant property by write-accessing it. The following FPScript, for instance, normalizes the data of a signal to 100%:

```
.UnitY = "%"; Signal / Maximum(Signal) * 100.
```

You should, however, only use this write access if the embedded FPScript formula transforms the data into a new unit and, in all other cases, ensure that the correct unit is entered in the data set accessed.

An embedded FPScript, however, is not used only for accessing data for presentation, but is also used to access attributes that are to be displayed as an axis label or in a table cell, for instance. To do this, use a formatter that has an embedded FPScript expression appended to it in the relevant text. This type of formatter always starts with a % sign and determines how the data is to be presented as well as from where the data originates. A table cell that displays a maximum value could, for instance, contain the following:

```
Maximum: %{Maximum(Signal)} %{Signal.UnitY}
```

Here, two fields are embedded in the text. The first field contains an FPScript expression that calculates the maximum of a signal, and the second field supplies the physical unit of the signal's Y component as a string. In the example, the field does not specify how the data is to be formatted. Fields that determine this use additional code between the % sign and the '{' curly bracket. The part of the field before the curly bracket is also called the **Formatter**.



By default, the following text is entered on the **Axis Labeling** tab for the axis of a 2D diagram:

```
%{.Curves(%Curve).Data.YValueObject.NameOrComments(.Curves(%Curve).Data.YComponent)} % {.Curves(%Curve).Data.YValueObject.UnitInBrackets(.Curves(%Curve).Data.YComponent)}
```

The first field provides the name or comments for the data set presented as a curve, and the second field provides its physical unit. Before the FPScript code is compiled, the % Curve placeholder is replaced by the number of the curve that is currently being processed. The two FPScript expressions use the diagram's Automation object model to access the attributes of the data object that is used for the curve's Y component. To be more precise, the embedded FPScript formula specified for the Y component of the curve is being accessed. This in turn usually takes the attributes from the data object to which it refers.

## 6.1 2D and 3D Diagrams

You can use diagrams to make attractive and expressive presentations of data and results of calculations. FlexPro offers you the Diagram Wizard: a powerful tool for creating a variety of diagram types. The wide variety of diagrams is the result of the flexible structure of just two object types: the 2D diagram and the 3D diagram.

The 2D diagram presents data in a rectangular two-dimensional plane, while the 3D diagram presents the data in a cube projected onto the plane. The 2D diagram is especially ideal for presenting signals and data series, and the 3D diagram is ideal for presenting signal series, data matrices and space curves. The diagram's plane, i.e. the rectangular area in which the curves are displayed, is surrounded by a border in which the axis labels and the legend are displayed. The size of the border's area can be adjusted.

For each data set that you want to display in a diagram, you have to add a curve to the diagram. This refers to the data and determines how the data is to be displayed. The variety of optional diagram types is the result of the large range of optional display formats coupled with their ability to be combined. For the curve of a 2D diagram, you can, for instance, combine any of the following display formats: connection line, symbols, error indicators, labeling, columns/bars, and fill.

Diagrams consist of the following elements, which are explained in the subsequent sections: axis, grid, curve, legend and color legend.

## Axis

The diagram plane or space is spanned by axes, which determine the scaling of the data to be displayed. The 2D diagram uses at a minimum one vertical Y axis and one horizontal X axis, and the 3D diagram also uses a Z axis, which is pointed toward the viewer. Please note that, contrary to the typical labeling in mechanics, the vertical axis in FlexPro is also called the Y axis in a 3D diagram. While 3D diagrams use a fixed number of three axes, one each for the X, Y, and Z directions, several Y and X axes can be used for a 2D diagram. This makes it possible to display curves with different physical units in the same diagram. If you use more than one Y axis, then you can alternatively stack them on top of each other.

## Axis Type

To scale the axes of 2D and 3D diagrams, you can use the linear, logarithmic with base 10, 2 or e, offset reciprocal, third octave, octave, probability, probit and logit scale types.

With **linear scaling** you can choose whether you would like to specify a fixed division interval or whether the division interval should be determined automatically so that there are a fixed number of divisions.

The **linear scaling with metric division** option lets you establish a fixed relationship between the length of an axis and the physical unit of the curve displayed against this axis. You can set the scale freely. For example, if you set 0.1 V/mm as the scale and 1 V as the division interval, the divisions of the axis will be exactly 1 cm apart. If you enlarge the diagram and the length of the axis changes accordingly, the number of divisions will increase automatically. The distance between the divisions will remain constant.

By selecting Linear, division by data set or Logarithmic, division by data set you can determine at which positions on the axis divisions are to be drawn by using a data set or a formula. The procedure used depends on the data set or formula that you specify for labeling:

- **Data series:** The values are taken from the data series in sequential order, and a division is drawn at the corresponding position. All numerical data types are allowed. The process ends when all values have been read.
- **Formula with one argument (function):** The function is called with an ascending index as the argument. The first index is a null value. The result of the function determines the position of the axis division. All numerical data types are allowed for the result. The process ends when a value lies outside of the starting or end value of the axis, or when a void floating point value or the value Empty is passed.

**Reciprocal scaling** uses the transformation  $X' = 1 / X$ .

Use **Offset reciprocal scaling** to convert from Kelvin to °C. The transformation is  $X' = 1 / (X + 273.15)$ .

**Probability scaling** is used to linearize normal distribution functions. If the Y axis of a diagram has been probability-scaled, these types of distribution functions appear as straight lines and can easily be compared.

**Probit scaling** corresponds to probability scaling, although here standard deviations are considered, where 50% corresponds to the value of 5. The value range of a probability or probit-scaled axis lies in the interval [0.0001%, 99.9999%].

**Logit scaling** is similar to probability scaling. As a transformation, however, the Logit function  $X' = X / (100 - X)$  is used. The value range of a logit-scaled axis is within the interval [0.1%, 99.9%].

The scaling types **Third octave** and **Octave** are logarithmic scalings with the divisions corresponding to the third octave or octave series.

### Unit

If you want to display several quantities, i.e. data sets with a physical unit as curves across an axis, then their units must be compatible. You can define an output unit for the axis. All curves are then converted to this unit before output. If you do not define an output unit, the axis is assigned the unit of the first curve whose data has a unit; all additional curves are transformed to this unit.

### Scaling

You can set the scaling of the axis manually by specifying the physical values that correspond to both ends of the axis. Alternatively, you can specify data sets or FScript formulas that provide end values. With **autoscaling**, these end values are automatically determined from the curves to be displayed. This means that the curves are displayed with optimal spread. If you select **Data minimum/maximum**, the extreme values of the curves displayed on the axis are used. By selecting **Data range lower limit/upper limit**, the data range values entered on the **General** tab of the data sets are used. You can exclude individual curves from autoscaling. Autoscaling is often used with the **Extend end values** attribute. This means that the extreme values found are not used directly as axis end values; instead, a correction takes place first, so that the end values of the axis come to lie on a division. The axis end values can be calculated dynamically using an FScript formula. In 2D diagrams with several axes, you can also link the end values of neighboring axes.

If all of the diagram axes have the same physical unit, you can achieve an undistorted picture using **isometric axis adjustment**. All of the axis end values are thereby corrected so that they correspond to the axis length ratios. The isometric axis adjustment means for example, that with a 2D diagram, circular curves are also displayed as circles and not as ellipses.

### Orientation

Usually the X axis is aligned from left to right, the Y axis is aligned from bottom to top and the Z axis is aligned from back to front. However, the orientation can be switched around for each axis. You can also add orientation arrows to the axes, which point in the direction of the increasing scalar values.

### Axis Origin

You can define an axis origin for all 2D diagram axes and for the Y axis of a 3D diagram. For 2D diagrams, the axis origin determines the point of intersection of the corresponding axis of the axis system unless this axis is positioned at the edge of the diagram. For a 3D diagram, the axis origin determines the location of the floor, unless it was positioned at one of the end values of the axis. This floor determines, for instance, the plane on which the 3D columns are placed and is displayed as a frame with divisions.

### Division

You can provide an axis with divisions and subdivisions that can be labeled with the values corresponding to their position on the axis.

The divisions are shown as small dashes on the axis, and you can change their type, length and color. The position of the divisions on the axis result from the axis scaling. You can either specify the amount that should correspond to one division interval or you specify the number of division intervals. It is also possible to specify the positions of the divisions as data series or, to calculate them using an FScript function, which obtains the index of the division to be calculated as the argument.

With the division origin, you determine the value from which the division is to start at the top and bottom. This does not have to lie between the starting value and end value of the axis. When the subdivisions are applied, the division intervals are split up into a certain number of subdivisions. You can specify this number. Logarithmic axes are treated separately: With a division of 1 and either a subdivision total of 2 or a subdivision total of 8 and the **Avoid overlapping** option selected, only positions 2 and 5 are labeled, for instance, " $10^2$  2 5  $10^3$ ".

The axes divisions and subdivisions, for which you have enabled the option **Account for this axis when drawing the grid** on the **Appearance** tab of the Properties dialog box, also determine the location of the grid lines, if a grid is to be displayed.

You can label the divisions and subdivisions of an axis with any values taken from a data set or calculated using a function. For labeling you can use a data series, a signal or a formula with an argument (function). FlexPro then labels the divisions and subdivisions as follows:

Data structure	Labeling	Typical use
Data Series	The values are taken from the data series one by one and written onto the axis. All data types, including strings, are permitted. The values have to be stored in the same order in the data set in which the labeling of the axis occurs. The division labeling then proceeds from the division origin to the end value and then to the starting value of the axis. This is why you should set the division origin to the starting value of the axis.	Labeling of the divisions with text originating from a data set.
Signal	The value resulting from the axis division is searched for in the X component of the signal and the corresponding Y value is then written onto the axis. If necessary, interpolation is performed.	Display signals against any measured quantity, which does not have to be monotone, e.g., time signals over a space coordinate.
Function	The value resulting from the axis division is passed to the function as an argument and the result of the function is written onto the axis. All data types are allowed for the result, including strings.	Linearization of any characteristic line.

The axis will continue to be scaled according to the scaling configured and, if applicable, the values established through automatic scaling.

## Axis Labeling

You can add an axis label to each axis of a diagram. You can specify the axis label text directly or use the text from the curves displayed on the axis, which also have an **Axis Labeling** tab in their Properties dialog box. You can enter any text in which you can embed Fields for accessing header information of the displayed data sets. FlexPro offers a variety of pre-defined fields for labeling axes so that you do not have to program these yourself.

If you display multiple curves and use their axis labels, then these form text stacked vertically written on the axis. However, you can turn this option off for individual curves to show the time information only once on the X axis, for instance. The axis labels of the individual curves can be colored in with the respective color of the curve to make label assignment easy.

The data that you display above an axis should all have the same physical unit. You can add this unit to the axis label, display it between the last two divisions or behind each division.

### Position

The axes of 2D diagrams can be displayed at either edge of the diagram, at both edges or in the axis origin of an assigned axis pointing in the other direction. Several Y axes in a 2D diagram can be displayed as stacked instead of side-by-side. Here, the diagram is divided into several areas, and you can adjust the size of these areas. It is also possible to limit the area across which an axis is stretched. You can, for instance, create a second Y axis that only stretches over an area of 0% to 10% of the height of the diagram to overlay a digital channel in a narrow band as a step line. You can also display several X axes side by side.

### Curve

A diagram can have any number of curves. Each curve refers to a data set to be displayed and describes how its data is to be presented visually.

A curve in a 2D diagram can show the scalar value, data series and signal data structures. Each curve is assigned to exactly one X axis and one Y axis. If you specify a data series, its data is used for the Y coordinates of the points of the curve and the X coordinates are generated automatically by numbering, beginning with zero. When the curve is drawn, the positions of the points in the diagram plane are determined by the axis scaling. Parts of the curve that lie outside the area of the diagram are clipped. Scalar values can be displayed as horizontal or vertical lines. If you show a two-dimensional data structure, e.g., a signal series, as a 2D diagram, then the Diagram Wizard creates one curve for each signal in the signal series. The 3D diagram is, however, preferred for presenting two-dimensional data structures, since it can automatically display signals of a signal series that are added later without having to add curves. Some display formats use additional data, e.g., a data set with error values to display error indicators.

A 3D diagram curve can show data matrix, signal series and space curve data structures. Analogous to the 2D diagram, X and/or Z components that are not present are also generated automatically by numbering, beginning with zero. Instead of a signal series or a data matrix, it is also possible to specify several signals or data series that are then managed as a list in the curve and together form a signal series and/or a data matrix. Unlike 2D diagrams, there is only one axis system in 3D diagrams, so all curves are applied to the same axes. For 3D curve types that require grided data, a space curve (scatter data) is automatically converted into a signal series before display. You can choose one of two algorithms on the **Data** tab to display the surface.

On the **Data** tab of the curve, not only can you enter a reference to a data set or a formula, but you can also enter any FPScript expression. For instance, use an index operation to present a signal segment or a particular signal from a signal series. The FlexPro Diagram Wizard uses FPScript expressions within curves, e.g., for displaying

percentages or decibels.

## 2D Curve Display Formats

The 2D diagram curves let you use the following display formats, which you can combine in any way you like:

- **Connection Line**

Use this display format to display signals. You can choose from several types of connection lines:

Type	Function
Straight line	Connects the points of the curve with a straight line.
Step line	Connects neighboring points with horizontal and vertical lines instead of diagonal lines.
Segment	Draws a line between every other point of the curve.
Triple segment	Connects two points of the curve with a line and then leaves a gap.
Spline	Connects the points of the curve with a spline curve. You can specify the number of sampling points to be calculated.
Compensating spline	Connects the points using a compensating spline curve. The spline curve does not necessarily run through the points. You can control this using a weighting factor.

You can specify the line style, color and arrow type. Another option is to use a color palette. You then have a multi-colored curve for displaying tolerance bands, for example.

If the data set contains void values, you can specify whether these are to be bridged with a line or whether a gap should be shown in the curve.

- **Symbols**

This display format draws a small symbol, e.g. a rhombus or a circle, at each point of the curve. You can freely set symbol shape, size and color. Alternatively, the size of the symbols can be determined using an additional data set. Particularly for the frequent occurrence of the "Symbols" plus "Connection Line" combination, you can specify that a small gap must be maintained between the symbol and the connection line. The connection line then stops just before the symbol.

The "Horizontal Line" and "Vertical Line" symbols are especially intended for adding range limits into the diagram. You can specify a scalar value that then determines the location of the line.

The "Vector" symbol is available for displaying vectorial measures. Here you can pre-define the angle and length of vectors or specify additional data sets.

- **Error Indicators**

The "Error Indicators" display format draws an error indicator at every point of the curve. The indicators can be drawn in Y or X direction and only in one or in both directions. The length of the error indicator from the point of the curve to the limiting line can be specified as an absolute value, as a distance to the point of the curve, or as a percentage value in relation to the point of the curve. To do this, you can enter fixed values or additional data sets.

- **Data Labeling**

The "Data Labeling" display format outputs a text with the value and unit of X and/or Y coordinate for the point at the location of each point of the curve. Alternatively, you can output values from an additional data set. The data set may contain numerical data or strings. You can freely set the writing direction and placement relative to the point.

- **Columns/Bars**

The "Columns" display format draws a column at each X value of the curve. This column runs from the X axis to the Y value. The "Bars" display format draws a bar at each Y value of the curve. This bar runs from the Y axis to the X value. These display formats are also available in stacked and differential variations as well as in a variant in which the base point is not the axis, but rather the axis origin. For stacked columns, the columns for several curves are displayed one on top of another instead of grouped side by side. Differential columns do not start at the X axis, but instead start with Y values taken from another data set. Analog applies to stacked bars and differential bars, but in the X direction in this case. You can set column or bar width, fill color and style.

With bar charts and column charts, you can either specify a fixed width as a percentage of the spacing between neighboring columns, or select the **Dynamic column width** option on the curve's **Columns/Bars** tab. In this case, the columns are drawn in a way that allows them to be seamlessly tagged onto each other. This is the case even if the X intervals between neighboring columns vary. The limit of neighboring columns always lies in the middle between two neighboring points of the curve.

- **Fill**

This display format fills the surface between the curve and the X or Y axis with a particular color, a hatching or a linear color gradient shading. A second data set can also be specified for differential filling.

Display formats for a single curve can be combined in any way you want; for instance,



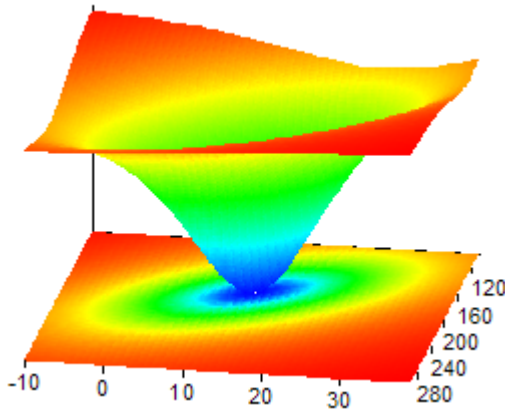
a connection line can be combined with symbols or error indicators. Alternatively, you can also set up several curves with different display formats; for instance, the first curve would have a connection line to represent a signal, and a second curve would have symbols to represent a data set that contains only local maxima in the signal as points.

### 3D Curve Types and Display Formats

As opposed to a 2D diagram, the following curve types differ for a 3D diagram:

- **Surface**

This type of curve displays the data as a surface defined by data points arranged in a grid. Space curves (scatter data or XYZ data) are automatically converted to a signal series (data grid) before display. The conversion takes place either through resorting or through "natural neighbor" interpolation. The surface can be smoothed, if necessary. For the "resort" algorithm, smoothing takes place later using bicubic splines. You can adjust all of the curve settings on the **Data** tab. You obtain a contour diagram by setting the diagram's angles of rotation and inclination so that the result is a plan view.



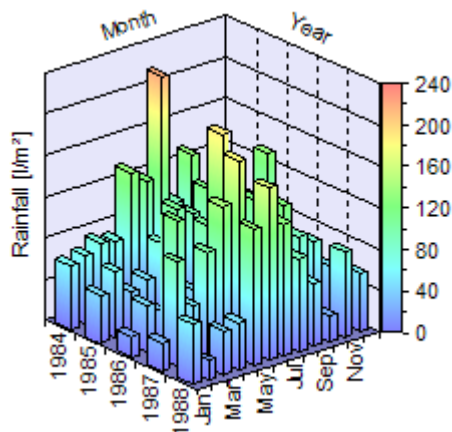
- **3D Columns**

You can display signal series (with a Z component, if applicable) or data matrices in a 3D column chart. A column chart can also be a combination of several data series or signals with identical X components.

You can specify a fixed width of 1 to 100 percent of the spacing between neighboring

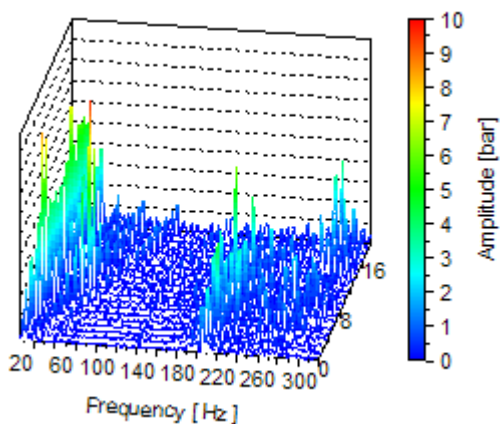
columns for the column width, defining the minimum distance in the X and Z direction to 100 percent. The spike chart means that columns are drawn as "needles", using the attributes from the **Fill** tab as the base. The border lines are omitted in this case.

When selecting **Color field matrix** in the Diagram Wizard, the result is a 3D column in plan view with a color shading or color palette in the Y direction.



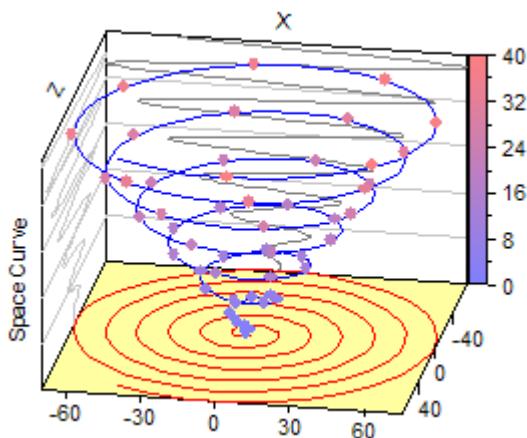
- **Waterfall**

In a waterfall chart you can display signal series (possibly with a Z component and two-dimensional X component), data matrices or space curves. A waterfall diagram can also be put together from several data series or signals. Space curves are converted automatically to a signal series before being displayed by resorting using the **CurveToSurface** function. You can change the tolerance for the conversion on the **Data** tab of the curve in the diagram.



- **Space Curve**

In a space curve chart, you can display data sets of the space curve data structure or three data series, which combine to form a space curve.



Usually, a 3D diagram only uses one curve from one of the four types. In some cases, however, it makes sense to use several curves, for instance, to label individual points in a contour chart.

## Display Formats

The following display formats can be used within a single type of curve:

- **Border Lines (3D Columns)**

To reinforce the impression of space, the borders of the three-dimensional columns can be highlighted with lines. These border lines are particularly important when no lighting source is used. Typical line attributes can also be applied to border lines.

- **Grid Lines (Surface)**

You can use grid lines to display the surface as a grid. Neighboring points are joined up by lines whose style and color you can set. It is possible to use a color palette in the Y direction. In addition to this, you can project the grid onto the coordinate system planes.

If you do not use any filling for the surface, you can also make the grid lines visible, although these are not visible from the viewing angle set.

If the data set contains void values, then all of the connection lines to these values are omitted, causing gaps in the grid network.

- **Connection Line (Space Curve, Waterfall)**

This display format connects a curve's adjacent points with a line for which you can adjust the style and color. For the waterfall chart, you can use a color palette in the Y or Z direction or linear or spectral color gradient in the Y direction. The space curve can be colored using a color palette in any direction.

If the data set contains invalid values, you can specify whether these are to be bridged with a line or whether a gap in the curve is to be shown.

The connection line for the waterfall chart can also be drawn as a step line, especially for displaying digital signals. You can freely choose the location of the vertical edge between neighboring points.

- **Symbols (Surface, Space Curve, Waterfall)**

This display format draws a small symbol, e.g. a rhombus or a circle, at each point of the curve. You can freely set symbol shape, size and color.

- **Data Labeling (Surface, Space Curve, Waterfall)**

The "Data Labeling" display format shows text with the value of the X, Y or Z coordinate of the point for each point on the curve. You can also use data from additional data sets for data labeling.

- **Fill (Surface, Waterfall, 3D Columns)**

This display format fills the surface, the area under each curve of a waterfall chart, or

3D column walls with a solid color, linear color gradient or a color spectrum. Alternatively, you can specify a palette with fill colors that can be used for the surface in the Y direction, for the waterfall chart in the Z direction, and for the columns in any direction. If desired, you can also project the filling of a surface onto the planes of the coordinates system.

If you use a color palette in the Y direction or one of the color shadings, you can switch on a color legend and thus visualize the color change regardless of the viewing angle of the 3D diagram. The coloring also depends on the lighting settings on the 3D diagram's **3D View**.

- **Contour Lines (Surface)**

These are intersection lines between the surface and planes, which are parallel to one of the three planes of the axis system. Contour lines emphasize the contour of a surface and can be drawn for any direction. The intersection planes can be specified by the divisions of the corresponding axis or by a freely selectable data set. The contour lines can also be optionally projected onto the corresponding plane of the coordinates system. You can also have a label with the value assigned to the contour line.

- **Projections (Space Curve)**

This display format projects the space curve onto the planes of the coordinates system. You can freely set the line attributes of the projection.

- **Perpendiculars (Space Curve)**

This display format allows you to draw perpendicular lines from the points of the curve onto the planes of the coordinates system. You can freely set the line attributes for the perpendicular lines.

## **Legend**

A diagram can have a legend that provides clear assignment of curves to the data on which the curves are based.

For each diagram curve selected to have a legend, a short section of the curve is drawn in the legend and labeled with text. For each curve, you can enter the text and insert fields that are replaced by header information from the curve's data set.

You can specify a title for the legend. You can color in the background, and a frame can be added to the legend. You can specify how many columns the legend is to have and in which order the curves are to be displayed. You can freely position the legend. It is, however, assigned to a corner of the diagram, so that when the diagram size is changed, the legend is shifted accordingly.

## Axis and Curve Labeling

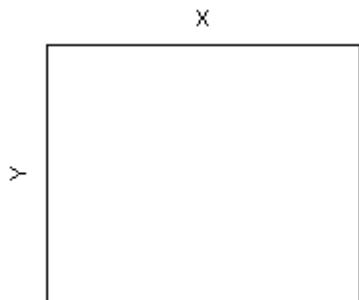
In addition to the legend, you can also use axis and curve labels to display particular data on the curve. The curve label is written directly on the course of the curve. You can set its position. You can specify the axis label for individual curves separately for each axis of the diagram. The lines of text are then written on the axis, with each line placed below the last line. You can add fields to the text to display attributes such as the name or unit of the data set upon which the curve is based.

## 3D View and Lighting

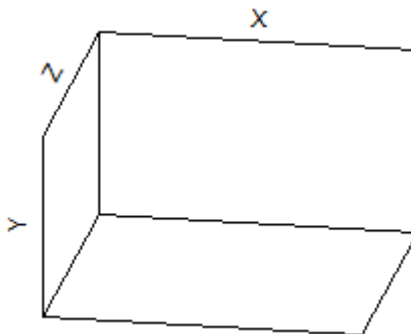
The 3D view is determined by specifying an angle of inclination and an angle of rotation. Here, the angle of rotation specifies by how many degrees the coordinates system should be rotated clockwise around the Y axis. The angle of inclination determines by how many degrees the coordinates system should be tilted counter-clockwise, relative to the projection plane.

An angle of rotation and an angle of inclination of 0 specify that the viewing angle is located exactly opposite the XY plane of the coordinates system.

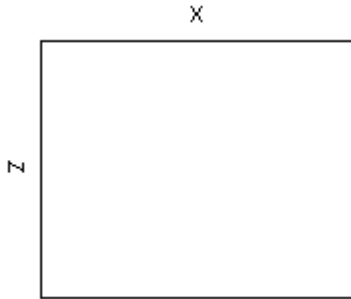
The following diagrams are a few examples:



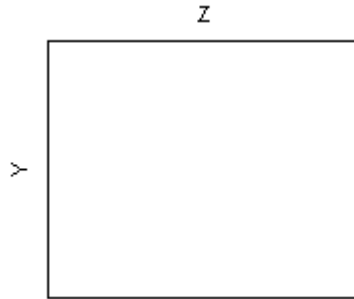
Angle of rotation = Angle of  
inclination =  $0^\circ$   
(opposite the XY plane, Z points to  
the viewer)



Angle of rotation =  $10^\circ$ , Angle of  
inclination =  $30^\circ$   
(Standard)

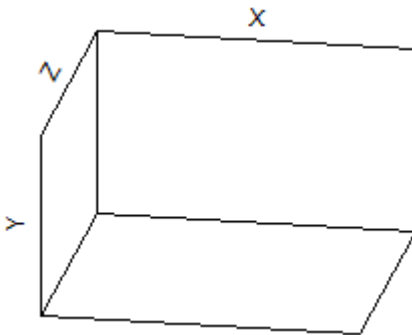


Angle of rotation =  $0^\circ$ , Angle of inclination =  $90^\circ$   
(plan view, Y points to the viewer)

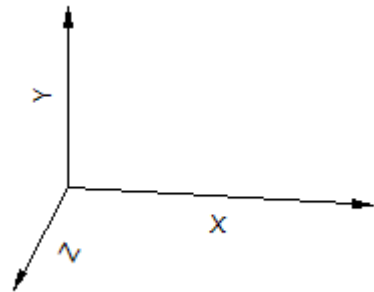


Angle of rotation =  $90^\circ$ , Angle of inclination =  $0^\circ$   
(opposite the YZ plane, X points to the viewer)

The axis system of 3D diagrams can be presented as an axis intercept with planes or as an axis intercept only:



Axes and planes



Only axes

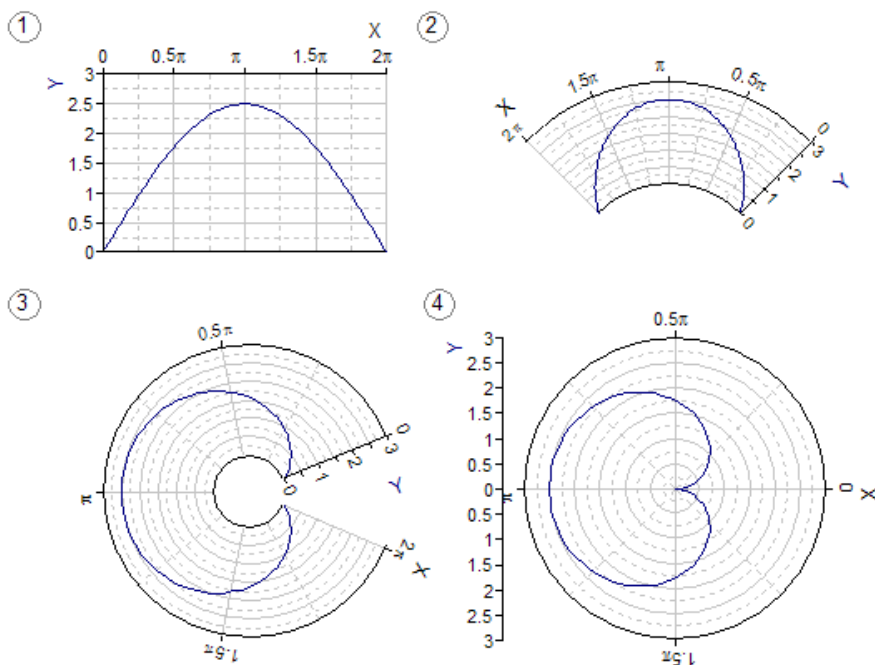
While the planes are always displayed behind the curve, the axis intercept is always displayed in the coordinates origin, even if it ends up in front of the curve.

For curves types with surfaces (surface, columns), you can also add a lighting source. The position of the lighting source is determined similarly to the viewing angle, but here, all of the angle specifications are relative to the viewing angle.

## Polar Graph

All 2D diagram display formats are also available for the polar graph. Here, the complete diagram is transformed into a full circle, a segment of a circle or a ring by using polar transformation. The X axis of the 2D diagram is transformed into an angle axis, and the Y axis is transformed into a radial axis.

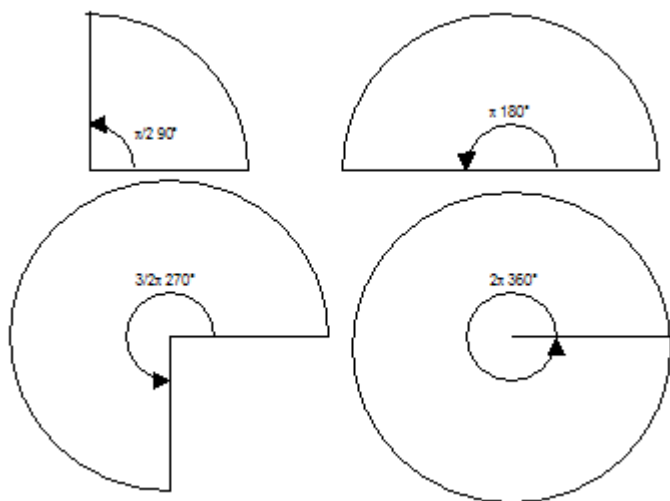
The following illustration shows the transformation from Cartesian graph to polar graph:



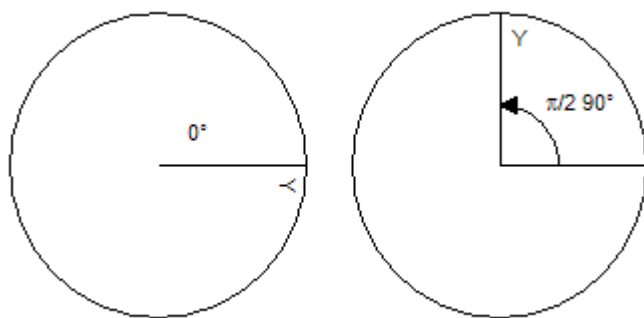
Note that the orientation of the X axis is reversed by the transformation.

On the **Polar Transformation** tab of the 2D diagram's Properties dialog box, you can activate polar transformation and set parameters for it. The transformation is controlled by the two opening angle and starting angle attributes, and inner radius. The **opening angle** determines the size of the circle segment upon which the diagram is formed. An opening angle of  $2\pi$  or  $360^\circ$  in this case is the equivalent of a complete circle.

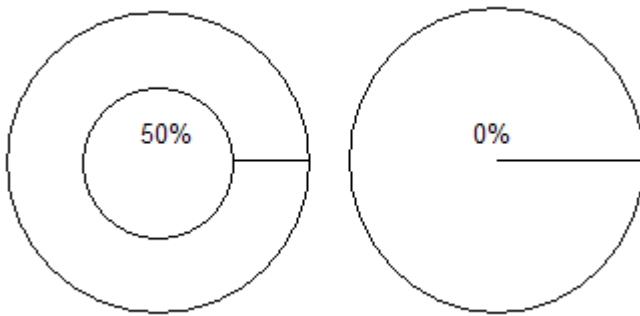




The **starting angle** determines the location of the diagram's original left border. The polar diagram can be rotated counterclockwise.



The **inner radius** is specified in percent and determines whether the diagram is to be in the shape of a circle or a ring.



The X values of the data to be plotted can include the angle in **radians** ( $0 - 2\pi$ ) or in **degrees** ( $0 - 360^\circ$ ). The diagram's angle axis is automatically labeled with the respective angles if this is set for metric axis division. It is also possible to label the angle axis in degrees, even though the measured values are in radians, and vice versa. Here, only the respective angle formatter for the axis must be specified.

A closed polar diagram no longer has a starting or end value for the angle axis. You can thus also plot data that stretches across a larger angular range than  $[0, 2\pi]$  or  $[0, 360^\circ]$ , i.e. over several "rotations". Also, with regard to positioning the Y axis or radial axis, the closed polar diagram deviates from the open polar diagram. While the Y axis, depending on the selected position, is plotted for an open diagram at one or both sides or within the circle segment, this is only displayed for the closed diagram within the circle if it was positioned at the axis origin of the X axis. Otherwise, the axis is positioned next to the full circle.

The diagram is usually an exact circle and fits into the center of the diagram's rectangular area. However, if you enable the option **Allow elliptical distortion**, then the diagram is distorted elliptically so that the diagram's plane is sized for best fit.

## Color Palette and Color Gradient

Using a color palette or color gradient can considerably improve legibility, especially of 3D graphics. In 2D diagrams, color palettes are often used for presenting ranges of values, such as tolerance bands and warning ranges. Color gradients, especially spectral color gradients, which simulate the colors of a rainbow, form the basis for what is called a 2 1/2D graphics contour and color field matrix. Here, the information for the third dimension is marked by the color.

The assignment of the colors for color gradient or for the color palette to the scaling the Y axis in a 3D diagram is represented by its color legend.

## Grid

You can provide the surface of the 2D diagram or the coordinates planes of the 3D diagram with a grid to increase the legibility of the diagram. The location of the grid lines is oriented toward the divisions and subdivisions of the axes spanning the plane. Since 2D diagrams can have any number of X and Y axes, each axis can be set separately, whether or not the grid lines are to be drawn for their divisions.

You can separately style the horizontal and vertical lines as well as the frame and the grid background. The frame and background are part of the grid in 2D diagrams only, since here the axes do not have to be positioned on the edges. In 3D diagrams, the framing of the planes as well as their background color are part of the axis system.

## Color Legend

In 3D diagrams, the color legend creates a relationship between the colors of the color shading used in a curve and/or a color palette and the scaling of the diagram's Y axis. This is another Y axis with a band of color, which is positioned next to the actual diagram.

A color legend uses the scaling of the diagram's Y axis. To do this, all settings must be applied at the Y axis. Otherwise, you can freely format the color legend as you would an axis.

## Working with Diagrams

### Creating a Diagram

1. In the Folders window, select the folder where the new diagram is to be placed.
2. Select the data you want to display as curves in a diagram. You can select data sets, formulas, analysis objects or a cell range in a folder's data view or in the window of an open data set. You can also use the range between the cursors of a diagram as data selection.

---

**Note:** If you want to display measurement data located in a subfolder, first activate it to show the measurement data sets.

---

3. If you want to embed the new diagram into a worksheet or document, at this time click on the open worksheet or document window.
4. Click on **Insert diagram** from the **FlexPro Objects** toolbar.
5. The Diagram Wizard appears to guide you through the remaining steps.

**Note:** You can easily create an embedded line diagram for a series of data sets by selecting the data sets in the Object List and using drag-and-drop to drag them into a document or worksheet.

---

## Resizing the Diagram Borders and Plane

### To resize the borders using your mouse:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a point on the plane of the diagram where there are no grid lines or curves.
3. Handles appear at the four borders of the diagram's plane. You can use these handles to resize the diagram's frame by dragging the borders with the mouse. You can resize the entire diagram by dragging the hatched frame surrounding it, provided the **Fit To Window** option in the **View** menu is switched off.

### To resize the borders and diagram plane, do the following:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on a part of the diagram's plane where there are no grid lines or curves.
3. Click on the **Size** tab.
4. Enter the desired measurements.

## Enabling and Disabling Automatic Margin Correction

FlexPro can automatically resize the borders of 2D and 3D diagrams if the diagram's elements extend beyond the edge of the window. You can enable or disable automatic margin correction by doing the following:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on a point on the diagram's surface where there is no curve.
3. Click on the **Size** tab.
4. Check the option **Correct margins automatically** or remove the checkmark.

### Changing the Background Color of the Diagram's Plane

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a point on the plane of the diagram where there are no grid lines or curves to select the diagram's plane. You can select the three planes of the coordinates system collectively in 3D diagrams by clicking on one of them while holding down the ALT button.
3. You can change the color for the fill by selecting a fill color on the **Drawing Tools** toolbar. To remove the fill, select **No Fill**.

---

**Note:** If you chose **No Fill** for the background color, you will not be able to select the background later by clicking on it. In this case, double-click on the diagram and set the desired color in the Properties dialog box.

---

### Toggling Grid Elements On and Off

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the diagram plane to open the Properties dialog box.
3. Click on the **Grid** tab.

#### 3D Diagram:

4. Select the tab for the coordinates plane for which you want to display a grid.
5. Select the grid elements to be displayed.

### Designing Grid Elements

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Select the grid elements for which you would like to change the style.
3. Set the line width and line style on the **Drawing Tools** toolbar.
4. The color of the grid elements can be set by selecting a line color from the **Drawing Tools** toolbar.

### Positioning the Color Legend of a 3D Diagram

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Use your mouse to select and drag the color legend to the desired position.
3. To specify which corner of the diagram you want to assign the legend, double-click on the surface of the color legend.
4. In the Color Legend Properties dialog box, select the **Position** tab.
5. Select the desired position in the field **To corner**.

### Splitting a 2D Diagram into Multiple Pages

1. If the diagram is in a document, double-click on the diagram to open it.
2. Use your right mouse button to click on the diagram and select **Properties** from the context menu.
3. Click on the **Page Splitting** tab.
4. Select the desired **Mode** for page splitting. Here, call up Help in the dialog box if necessary.
5. In the field that now appears, specify the desired **Number of pages** or **Number of divisions per page**.

### Navigating Multi-Page 2D Diagrams

- Use the commands **Previous Page** or **Next Page** from the **View** menu or **Drawing Tools** toolbar to go to the previous or next page of the diagram.
- To show the first or last page of the diagram, use the commands **First Page** or **Last Page** from the **View** menu.
- You can use the command **Choose Page** from the **View** menu to select any page.
- You can use the mouse wheel to scroll through several pages of the diagram.

### Changing the 3D Diagram Viewing Angle

**To change the viewing angle, do the following:**

1. On the **3D View** toolbar, click on **Rotate Right** to rotate the diagram clockwise by 10°. You can rotate the diagram by 10° in the opposite direction by clicking on **Rotate Left**.

2. Click on **Increase Inclination** to increase the angle of inclination by 10°, or on **Decrease Inclination** to decrease it by 10°.
3. Click on **Plan View** to view the diagram from above. The Y axis is now pointing toward you and is no longer visible. The same applies to the Z axis if you click on **Front View** and for the X axis if you click on **Side View**.
4. With **Standard View** you can set the view back to the default

**If you want to set the angles more precisely, do the following:**

1. Use the right mouse button to click on the diagram and select the **Properties** command from the context menu.
2. Select the **3D View** tab in the dialog box displayed.
3. In the **Perspective** box, select one of the viewing angles provided or select **Free position** and enter the angle.

### **Changing the 3D Diagram Lighting Parameters**

To change the lighting parameters for a column or surface diagram, do the following:

1. Use the right mouse button to click on the diagram and select the **Properties** from the context menu.
2. In the dialog box that appears, select the **3D View** tab.
3. Turn on the lighting, select the desired lighting position and set the contrast.

### **Embedding Graphics**

1. Choose the command **Insert Image** from the **Diagram** menu.
2. Choose a file in the **Insert Image** dialog box.
3. Click OK to close the dialog box. The newly inserted image is now displayed.
4. If necessary, drag the image with your mouse to position it.
5. You can change the absolute height and width as well as the relative size in relation to the original image size in FlexPro's Properties window.

## Working with Axes

### Adding or Removing an Axis

#### To remove an axis:

1. Select the axis that you would like to delete.
2. Press the DEL key.

---

**Note:** You cannot delete the last Y axis or X axis.

---

#### To add an axis:

1. Select an existing axis of the same type (X or Y axis).
2. Use your mouse to drag a copy of the axis to the desired position.

### Specifying the Output Unit

1. Double-click on the axis.
2. In the Properties dialog box click on the **Scaling** tab.
3. Under **Unit** enter the desired output unit or choose the unit from the list.
4. Click **OK** to close the dialog box.

---

**Note:** If you specify an output unit, the unit of the data displayed across this axis must be compatible.

---

### Changing the Axis Origin

1. Double-click on the axis.
2. In the axis Properties dialog box, click on the **Scaling** tab.
3. Enter the axis origin in the field of the same name. The axis origin can be set to zero, but does not have to be.

---

**Note:** Only the Y axis in 3D diagrams has an axis origin.

---



### Positioning an Axis in a 2D Diagram

1. Double-click on the axis with the position you would like to change.
2. In the axis Properties dialog box, click on the **Position** tab.
3. Now choose in the **Position** field whether the axis is to appear on the left or right of the diagram or in the axis origin of the axis going the other direction. If the axis is a further axis at the same position, you can also specify the **spacing to the previous axis**. If you chose **In axis origin**, then specify the axis origin of the axis you want to use in the **Assigned axis** field.

---

**Note:** You can use your mouse to position adjacent axes. To do this, click on the axis and position it by dragging it with the mouse.

---

### Stacking the Y Axes of a 2D Diagram

To stack the Y axes of a 2D diagram, do the following:

1. Double-click on the area of the diagram, but not on the curve.
2. In the diagram Properties dialog box, click on the **Axes** tab.
3. For **layout of Y axes**, select **Tile vertically**.

### Adjusting the Height of Stacked Y Axes

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.

#### Using the mouse:

2. Drag the dividing line between the two diagram planes to the desired position.

#### Using the Properties window:

2. Select the Y axis whose height you want to adjust.
3. In the **Size percentage** field of the Properties window, specify the desired height of the axis relative to the overall height of the diagram.

### Displaying the X Axes of a 2D Diagram Side By Side

FlexPro cannot stack X axes of a diagram like it can Y axes, but you can obtain the same effect by doing the following:

1. If the diagram does not have two X axes, first add an axis.
2. Double-click on the diagram's first X axis.
3. In the axis Properties dialog box, click on the **Position** tab.
4. In the **Start of axis** field, enter **0%** and in the **End of axis** field, enter **50%**.
5. Click on OK to close the dialog box.
6. Double-click on the second axis and switch again to the **Position** tab.
7. In the **Position** field, select the same position that you used for the first axis.
8. In the **Spacing to the previous axis** field, enter **0**.
9. In the **Start of axis** field, enter **50%** and in the **End of axis** field, enter **100%**.
10. Switch to the **Appearance** tab and make sure that the option **Account for this axis when drawing the grid** is checked.

### Adding or Deleting an Axis Division

#### To delete an axis division:

1. Double-click on the axis division.
2. On the **Appearance** tab of the Properties dialog box of the axis or color legend, uncheck the **Divisions** or **Subdivisions** checkbox.


#### To add an axis division:

1. Double-click on the axis to which you would like to add a division.
2. In the axis Properties dialog box, click on the **Appearance** tab.
3. Check the **Divisions** and/or **Subdivisions** checkbox.

### Changing Division Length and Type

1. Select the division that you would like to edit. You can simultaneously select the divisions for several axes and the color legend by holding down the CTRL key. Select all divisions of the diagram by holding down the ALT key.
2. You change the length of the division dashes by specifying a different value in the

**Size** field on the **Position** toolbar.

3. You can change the division type by clicking on the corresponding icon in the **Division** menu on the **Position** toolbar. For instance, click on  to set the divisions of an X axis to **Division lines in both directions**.

### **Toggling Isometric Axis Adjustment On and Off**

1. Double-click on a point in the diagram plane where there is no curve.

#### **2D Diagram:**

2. In the diagram Properties dialog box, click on the **Axes** tab.
3. Check the option **Adjust primary system of axes isometrically** or remove the checkmark.

#### **3D Diagram:**

2. In the diagram Properties dialog box, click on the Axis System tab.
3. Check the option **Adjusting axes isometrically** or remove the checkmark.

### **Adding or Deleting a Division Label**

#### **To delete a division label, do the following:**

1. Double-click on the division label.
2. On the **Division Labeling** tab of the Properties dialog box of the axis or color legend, uncheck the **Division labeling on** checkbox. If you would only like to switch off the subdivision labels, remove the checkmark from **Also label the subdivisions**.

#### **To add a division label, do the following:**

1. Double-click on the axis or color legend to which you would like to add a division label.
2. In the Properties dialog box, click on the **Division Labeling** tab.
3. Check the **Division Labeling** checkbox and, if applicable, **Also label the subdivisions**.

## Changing the Division Label Format

**To change the formatting of all values of an axis division label, do the following:**

1. Double-click on any division label with a format that you would like to change.
2. Click on the button on the right side of the **Format** input field to adjust the formatter.
3. Select the desired format.

**Using the Properties window, you can set formatting for several axes at once:**


1. If the Properties window is not already open, click on **View** in the menu bar, then point to **Window** and then click on **Properties**.
2. Select the division labels for several axes whose formatting you would like to change by clicking on these while holding down the CTRL key. Select all diagram division labels by holding down the ALT key.
3. Now click in the **Format** input field of the Properties window and then click on the button that appears to the right of the field.
4. Select the desired format.

---

**Note:** The formatting applies to all of the division labels for the selected axes.

---

## Changing the Position and Alignment of the Division Label

1. Click on the division label to select it.
2. On the **Position** toolbar, click on the appropriate symbol in the **Position** menu to select the position. For instance, click on  to position the label at the top left of the division marker.

## Labeling the Axis Divisions with Text

1. Double-click on the axis with the division that you would like to label.
  2. In the Properties dialog box, click on the **Division Labeling** tab.
  3. Select **Division labeling using a data series, signal or function**.
  4. Specify a data series with strings that contains the desired text, or specify directly an FScript expression that generates the data series, such as {"One", "Two", "Three"}.
- The values must be specified in the same order as the axis labeling. The division

labeling then occurs from the division origin to the end value and then to the starting value of the axis.

### Displaying the Unit in the Axis Division

1. Double-click on the axis division label in which you want to add the unit.
2. On the **Division Labeling** tab, select the option **Add unit to division labeling**.
3. In the field under this option, choose whether you want to display the unit **at every division** or **between the last and second to last division**.

### Adding or Deleting an Orientation Arrow

#### To delete an orientation arrow, do the following:

1. Double-click on the orientation arrow.
2. On the **Appearance** tab of the axis or color legend Properties dialog box, uncheck the **Orientation arrow** checkbox.

#### To add an orientation arrow, do the following:

1. Double-click on the axis or color legend to which you would like to add an orientation arrow.
2. In the Properties dialog box, click on the **Appearance** tab.
3. Check the **Orientation arrow** box and specify the desired **Position**.

### Positioning an Orientation Arrow

1. Double-click on the orientation arrow.
2. On the **Appearance** tab of the axis or color legend Properties dialog box, select the desired position under **Orientation arrow**.
3. Under **Length**, enter the desired length of the arrow.
4. Under **Spacing**, specify, if applicable, the distance from the division label or from the axis label.

---

**Note:** Positioning the orientation arrow on the axis label of 3D diagrams is only an option when **Parallel to the axis** is set as the **Angle** on the **Axis Labeling** tab.

---

## Displaying an Axis with Customized Scaling

FlexPro can label the divisions of an axis with values that can be calculated using a function. This allows you to develop your own scaling. The following example shows how you can display a square characteristic curve as linearized, for instance. The characteristic curve is linearized using its inverse function, in this example the square root function, and displayed in a linearly scaled diagram. The Y axis of the diagram is labeled using the square function.

Do the following:

1. Create an FScript function called **Linearized** that allows you to linearize the data set with the characteristic curve. The formula must contain the following:

```
Arguments  x
          Sqrt(x)
```

2. Use the mouse to drag the **CharacteristicCurve** data set onto this formula in order to obtain the linearized characteristic curve as new formula called **CharacteristicCurveLinearized**.

3. Create an FScript function called **Parabola** used to square the division labels of the Y axis:

```
Arguments  x
          x^2
```

4. Select the formula **CharacteristicCurveLinearized** and call up the Diagram Wizard to create a linearly scaled 2D diagram. The characteristic curve is to be displayed as a straight line. The divisions of the Y axis, however, still correspond to the linearized values.
5. Double-click on the division label of the Y axis to open the **Division Labeling** tab.
6. Check the option **Division labeling using a data series, signal or function**.
7. Check the option **Also for data labels, markers and in the coordinates window** to make sure that the transformation will also be used there.
8. Enter the function **Parabola** into the field for the data set.
9. After you have exited the dialog box by clicking on OK, the squared values will be displayed on the Y axis.

**Changing the Axis Scaling**

1. Double-click on the axis.
2. In the axis Properties dialog box, click on the **Scaling** tab.
3. Select the desired scaling options.

**Changing the Number of Subdivisions**

1. Double-click on the axis line.
2. On the **Scaling** tab of the axis Properties dialog box, enter the number of subdivisions per division in the field with the same name.

**Adding or Deleting an Axis Label****To delete an axis label, do the following:**

1. Double-click on the axis label that you want to delete. If it is the axis label of a curve, the curve Properties dialog box appears. Otherwise, the axis Properties dialog box will appear.
2. On the **Axis Labeling** tab of the curve, uncheck the option **Add curve to Y , X or Z axis label**. Otherwise, uncheck the option **Axis labeling on**.


**To add an axis label to a curve, do the following:**

1. Double-click on the curve to which you want to add the axis label.
2. In the dialog box, select the **Axis Labeling** tab.
3. Select one or more of the options **Add curve to Y, X or Z axis label**.
4. Click on the **Y, X or Z axis properties** and make sure that the option **Use axis labels from curves** is selected.

**To write an axis label directly on an axis, do the following:**

1. Double-click on the line or division of the axis to open its properties dialog box.
5. Switch to the **Axis Labeling** tab.
6. If applicable, check the option **Axis labeling on**.
7. Uncheck the option **Use axis labels from curves**.
8. Now enter the desired label in the **Text** field.

### Changing the Position and Alignment of the Axis Label

1. Click on the axis label to select it.
2. On the **Position** toolbar, click on the appropriate symbol to choose the position. For instance, click on  to position the label at the top left of the axis.
3. In the **Angle** field, choose the writing angle at which the text is to appear.
4. If necessary, use the mouse to move the axis label in order to correct the label's position.

### Displaying Data Set Information in the Axis Label

1. Double click on the curve axis label in which you want to insert header information from the data set. If necessary, you may need to insert it first.
2. In the **Text** field, place the cursor at the position where you would like to insert a field.
3. Select the desired item from the **Fields** list box. A field is then inserted into the text. This field is later replaced by the updated data set information.

### Displaying the 3D Axis System as an Axis Intercept

1. Use the right mouse button to click on the diagram and select the **Properties** command from the context menu.
2. In the dialog box that appears, select the **Axis System** tab.
3. Under **Visualization**, select the option **Only axes** and click **OK** to close the dialog box.

### Labeling Only the Axis End Values

1. Double-click on the axis.
2. In the Properties dialog box of the axis, click on the **Scaling** tab.
3. As the **Axis type**, select **Linear, number of divisions**.
4. Select the option **Extend end values**.
5. In the **Division origin** list box, select **equal to start value**.
6. As the **Number of division intervals** enter 1.



**Note:** If you want the axis to be labeled with the exact minimum and maximum of the curve, you will need to select as the **Scale start value/end value** the **Data minimum/maximum** and enter the value 0 in the **Significant digits** field.

---

## Working with Curves

### Switching On a Display Format

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve for which you would like to toggle the display formats or on the curve's axis label.
3. In the dialog box that appears, click on the tab corresponding to the display format.
4. Select the option corresponding to the display format.

---

**Note:** You can combine several display formats with each other, such as **Connection line** with **Fill**.

---

## Rearranging Curves

### Using the diagram Properties dialog box:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.

#### 2D Diagram:

2. Use your right mouse button to click on the diagram. Make sure that nothing in the diagram has been selected.
3. Click on **Properties** in the context menu.
4. In the dialog box that appears, click on the **Curves** tab.
5. Select a curve from the **Curves** list.
6. Move the curve using the **In Foreground** or **In Background** buttons.

#### 3D diagram displaying several data series or signals:

2. Double-click on the curve.
3. In the curve Properties dialog box that appears, click on the **Data** tab.

4. Select a data set in the list.
5. Move the data set to the background using the **Up** button or use the **Down** button to move it to the foreground.

### **In the 2D diagram using the Properties window:**

1. Select the curve that you want to send farther back or bring forward, or select its axis label.
2. Enter a different value in the **Plot order** field of the Properties dialog box. Decreasing the value shown, sends the curve farther back. Increasing the value brings it forward.

### **Adding a Curve to a 2D Diagram**

The easiest way to add a curve to a 2D diagram is to drag the data set to be displayed as the curve with the mouse from the Object List into the diagram. If you use several Y axes, you have to drop the data set onto the Y axis that is to be assigned to the curve.

However, you can also insert an empty curve and then enter the data set manually:

1. Use your right mouse button to click on the diagram. If the diagram is in a worksheet or document, then you first have to activate it by double-clicking on it.
2. Click on **Insert Curve** from the context menu.
3. In the dialog box that then appears, enter the data set for the curve. You can also specify separate data sets for the X and Y coordinates of the points of the curve.

### **Changing the Axis Assignment for a 2D Curve**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve with the axis assignment that you would like to change, or click on its axis label.
3. In the dialog box that appears, click on the **Axes** tab.
4. Select the desired Y axis and/or X axis from the list boxes.

### **Excluding a Curve from Axis Autoscaling**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label that is to be disregarded when autoscaling.

3. In the dialog box that appears, click on the **General** tab.
4. Uncheck the option **Account for this curve when autoscaling the axes**.

## Hiding Curves

### Using the Properties dialog box:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the diagram, but not on the curve.
3. In the diagram properties dialog box that appears, click on the **Curves**.
4. In the **Visible** column of the **Curves** list, remove the checkmark from all curves that you want to hide.

---

**Note:** You should use the **Delete Curve** button to remove curves from the list that you want to hide permanently.

---

### Using the Properties window:

1. Select a curve by clicking on the curve itself or on its axis label. You can select several curves at the same time by holding down the CTRL key and clicking on them. Select all diagram curves by holding down the ALT key.
2. In the Properties window under **Visible**, select **False**.

## Opening the Data Set of a Curve

1. Select the curve whose data set you want to open by clicking on the curve or on its axis label.
2. Hold down the right mouse button to open the context menu and choose the command **Open Data Set**.

### Adding a Curve Label

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or the axis label of the curve to which you want to add a curve label.
3. In the dialog box that appears, click on the **Curve Labeling** tab.
4. Check the option **Label curve**.
5. Enter the **text** with which you want to label the curve. You can insert fields into the text to include attributes of the data set upon which the curve is based.
6. Specify under **Location** the relative location of the label to the course of the curve.

---

**Note:** You can adjust the location of the cursor label later using your mouse.

---

### Positioning a Curve Label

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Select the curve label that you would like to position.
3. On the **Position** toolbar, choose an ideal spot for placing the label above or below the curve.
4. Drag the text to the desired position on the curve.

---

**Note:** You can move the curve label only along the curve. For 3D diagrams, it is best to choose the front view or plan view.

---

### Adding a Data Series to a 3D Curve

To add a data series to a 3D curve that displays several data series or signals, do the following:

1. Drag a signal or a data series from the **Object List** onto a 3D diagram.
2. A question appears, asking whether you would like to add the data set to the last curve. Answer this question by clicking on **Yes**.

3. To add more signals or data series to the graphic, repeat this step.

---

**Note:** Any signals that you add to a 3D diagram in the manner described above must have an identical X component or none.

---

### Displaying Digital Data as a Step Line

You can set the appearance for a 2D line diagram and 3D waterfall chart as a step line:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or on the axis label of the curve for which you would like to add step lines.
3. In the dialog box that appears, click on the **Connection Line** tab.
4. For a 2D diagram, select the **Connection line on** option and select **Step line** from the **Type** field.  
For a 3D diagram, select the options **Connection line on** and **As step line**.
5. In the **Location of the vertical line** field, specify where the step is to lie between neighboring points.

### Displaying the Hidden Grid Lines of a Surface

1. If the 3D diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the surfaces for which you would like to turn the hidden lines on or off.
3. In the dialog box that appears, click on the **Lines** tab.
4. Check the option **Hidden lines**.

---

**Note:** If you have surface filling turned on, the **Hidden lines** option will have no effect.

---

### Smoothing a Surface

You can smooth the surface using bicubic spline interpolation.

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.

2. Double-click on the surface you would like to smooth.
3. In the dialog box that appears, click on the **Data** tab.
4. Specify the smoothing factor for **Points per X and Z interval**. This applies to both space directions.

---

**Note:** The display time increases quadratically with the smoothing factor.

---

### Designing Columns/Bars

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Select the columns that you would like to style.
3. Select the desired line width for the border and the desired fill style for 2D columns from the **Drawing Tools** toolbar.
4. You can change the color for the border and filling by selecting a line or fill color on the **Drawing Tools** toolbar.

### Changing the Column/Bar Width

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on any column or bar in the curve.

#### 2D Diagram:

3. In the **Column width** field, choose either **Fixed** and specify the value in percentage of distance from the neighboring columns, or choose **Dynamic** to set the columns seamlessly side-by-side for a different X interval.

#### 3D Diagram:

3. Switch to the **Display** tab and choose either **Column with width of** and specify the width, or choose **Spike graphic**.

### Changing the Bottom Line of Columns/Bars

The bases of columns and bars of 2D curves can either be on the other axis or on the axis origin of the axis in whose direction they are drawn. This is the Y axis for columns and the X axis for bars.

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on any column or bar in the curve.
3. Under **Type**, select a column or bar type with the desired bottom line.

### Adding Contour Lines in One Direction

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the surface for which you would like to add a contour line.
3. In the dialog box that appears, click on the **Contour Lines** tab.
4. Select the tab that corresponds to the direction for which you would like to add contour lines.
5. Check the option **Contour lines on**.
6. If you want to label the contour lines, select the option **Labeling on**.

### Editing Contour Lines

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the surface with the contour lines that you would like to edit.
3. In the dialog box that appears, click on the **Contour Lines** tab.
4. Select the tab with the contour line type that you would like to edit.
5. Make the desired changes in the corresponding fields.

### Adding Contour Line Labels

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the surface with the contour lines that you would like to edit.
3. In the dialog box that appears, click on the **Contour Lines** tab.

4. Select the tab for the contour line type for which you would like to add the label.
5. Check the option **Labeling on**.

### **Adding a Color Ramp**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click the fill that you would like to set for the color ramp.

**2D Columns:** On the **Columns/Bars** tab, select the type of **Color Gradient** from the **Fill** field.

**2D Surface:** On the **Fill** tab, select **Color Gradient** from the **Fill Style** field.

**3D Column, 3D Surface, 3D Waterfall:** On the **Fill** tab, select **Color Gradient** or **Color Spectrum** from the **Fill Style** field.

**3D Waterfall:** To fill the area under the curves with a color ramp, click on the **Fill** tab and select **Color Gradient** or **Color Spectrum** from the **Fill Style** field. **To add a color ramp to the lines of a curve, select Color gradient or Color spectrum from the Coloring field on the Connection Line tab.**

**3D Space Curve:** To add a color ramp to the symbols, click on the **Symbols** tab, select the option **Fill** and select **Color Gradient fill style** or **Color Spectrum**.

### **Changing the Fill Direction for a 2D Curve**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the filling.
3. Set the desired filling direction in the field with the same name.

---

**Note:** If you use the fill direction **Area between curve and Y axis** or **Area between curve and origin of X axis**, you should select the option **Swap Y and X components** from the **Data tab of the curve's** Properties dialog box . The curve is then no longer drawn from left to right, but from bottom to top instead, which matches this filling direction.

---



### Changing a Curve's Fill Color and Style

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Select the filling whose design you would like to change. You can select several fillings at the same time by holding down the CTRL key and clicking on them. You can select all filling by pressing the ALT key.
3. You can change the color for the filling by selecting a fill color on the **Drawing Tools** toolbar. To remove the filling, select **No Fill**.
4. For a 2D diagram, select the desired fill style from the **Drawing Tools** toolbar.

### Changing the Indicator Size

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on any error indicator of the curve. You can select several curves at the same time by holding down the CTRL key and clicking on them. You can select all curves by pressing the ALT key.
3. Set the size of the vertical line terminating the indicator on the **Position** toolbar.

### Changing the Indicator Type

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label.
3. In the Properties dialog box of the curve you can set the indicator type in the **Type** drop down list on the **Error Indicators** tab.

### Changing the Number of Indicators

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on any error indicator of the curve to open its Properties dialog box.
3. In the **Start at** field of the **Error Indicators** tab, enter the index of the first data point for which an error indicator is to be displayed. In the **Spacing** field, specify a spacing for each data point where an error indicator is to be displayed.

### Selecting the Data to Label

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label.
3. Click on the **Data Labeling** tab in the Properties dialog box.
4. In the **Fields** list, select the items of the data points to be used for labeling.
5. If you selected the placeholder **%<DataSetValue>**, then specify the name of the data set in the **Data Set** field from which the data to be labeled should originate.
6. In the **Text** field, arrange the fields as desired and, if necessary, enter additional text that you want displayed in each data label.
7. In the **Start at** field, enter the index of the first data point for which a label is to be added. In the **Spacing** field, specify a spacing for each data point where a label is to be displayed.

### Changing Data Label Formatting

#### To change the formatting of all values of a curve labeling, do the following:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on any value of the labeling whose format you would like to change.
3. On the **Data Labeling** tab of the Properties dialog box of the curve, set the insertion point in the **Text** text box on the field whose formatting you want to change. An example would be "%<YValue>".
4. Next, click on the button **Edit Formatter**.

#### With the FlexPro Properties window, you can set formatting for several curves simultaneously:

1. If the Properties window is not already open, click on **View** in the menu bar, point to **Window** and then click on **Properties**.
2. Select all curves with formatting that you would like to change by clicking on the curves while holding down the CTRL key. You can select all curves by pressing the ALT key.
3. Now in the Properties window, click in the **Text** text box under **Label** and then click on the button that appears to the right of the box.

4. In the dialog box that now appears, click on the **Edit Formatter** button.

---

**Note:** The formatting applies to all of the values in the curve selected.

---

### Changing Data Label Text Position and Direction

1. Click on the curve data labeling to select it. You can select several curves at the same time by holding down the CTRL key and clicking on them. You can select all curves by pressing the ALT key.
2. On the **Position** toolbar, click on the appropriate symbol to choose the position. For instance, click on **Position Top Left** to position the label at the top left corner of the curve point.
3. Choose the desired writing angle from the second drop down menu of the **Position** toolbar.

### Creating Lines

1. If the diagram is in a document or a worksheet, double-click on the diagram to activate it.
2. Select the lines in the curve that you want to change; for instance, this might be the connection line of a curve, the grid line of a surface chart, or the border of columns. Press the CTRL key to select several items. Press the ALT key to select all items of a particular type.
3. The color of the border lines can be set by selecting a line color from the **Drawing Tools** toolbar. To remove a connection or grid line, select **No Color**. You can change the line style and width using the appropriate icons on the **Drawing Tools** toolbar. Change the transparency using **Line Transparency** from the **Selection** menu.

---

### Notes:

A line style cannot be selected for column borders.

The border around 2D columns cannot be turned off.

The border lines of individual columns cannot be changed because only the entire curve can be selected.

---

### Changing Display of Void Values

Points in the curve, where either the Y component or the X component is void, cannot be drawn. If you use the **Connection Line** display format, you can specify whether void points are to be bridged or whether they should appear as gaps in the curve.

Do the following:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or on the axis label of the curve for which you would like to adjust the settings.
3. In the dialog box that appears, click on the **Connection Line** tab.
4. Check or uncheck the **Bridge void values** option.

### Turning On a Color Palette

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve for which you would like to use a color palette.

### 2D Line Diagram:

3. In the dialog box that appears, click on the **Connection Line**.
4. Check the option **Palette on**.
5. In the **Direction** field, specify whether the color is to change in the X direction or Y direction.

### 3D Waterfall Diagram:

3. In the dialog box that appears, click on the **Connection Line**.
4. Select **Palette** from the **Coloring** drop down list.
5. In the **Direction** field, specify whether the color is to change in the X direction or Z direction.

### 3D Surface with Grid:

3. In the dialog box that appears, click on the **Grid Lines** tab.
4. Check the option **Y Palette**.

**3D Columns:**

3. In the dialog box that appears, click on the **Fill** tab.
4. In the **Fill Style** drop down list, select **Color Palette**.
5. In the **Direction** field, specify whether the color is to change in the X, Y or Z direction.

**Connection Line for a 3D Space Curve:**

3. In the dialog box that appears, click on the **Connection Line** tab.
4. Check the option **Palette on**.
5. In the **Direction** field, specify whether the color is to change in the X, Y or Z direction.

**Symbols for a 3D Space Curve:**

3. In the dialog box that appears, click on the **Symbols** tab.
4. In the **Fill Style** drop down list, select **Palette in Y direction**.

**Determining the Interval Limits for a Color Palette**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label with the color palette interval limits that you would like to change.
3. In the dialog box that appears, click on the **Connection Line**, **Grid Lines** or **Fill** tab.
4. In the **Form palette intervals with** field, specify whether the palette intervals are to be predefined by the divisions of the corresponding axis or by a data set. Linking the palette to the axis division is only possible with linear or logarithmic scaling.
5. If you would like to specify a data set, enter its name in the **Data Set** field. You can also enter an FScript expression that calculates a data series directly in the field.

---

**Note:** Since the two outer intervals are open intervals, the data set must contain only n-1 interval limits for n intervals.

---

### Editing a Color Palette

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label with the color palette that you would like to edit.
3. In the dialog box that appears, click on the **Connection Line** or **Grid Lines** tab.

### Adding a Palette Entry

1. Click **Paste**.
2. Click on **Color**, **Line style** or **Line width** of the new entry to set the attribute.

### Moving a Palette Entry

1. In the list, select the entry which you would like to move.
2. Click **Up** or **Down**.

### Changing a Palette Entry

1. Click on **Color**, **Line style** or **Line width** of an entry to change its attribute.

### Deleting a Palette Entry

1. In the list, you select the entry which you would like to delete.
2. Click **Delete**.

### Transferring a Palette from One Diagram to Another

1. Open the tab of the diagram with the palette that you want to transfer, as described above.
2. Click **Save**.
3. Select the saving location, enter a file name, and click **OK**.
4. Close the **Properties** dialog box by clicking **Cancel**.
5. Open the tab of the diagram to which you want to assign the palette, as described above.
6. Click **Load**.
7. Select the previously saved database and then click **OK**.

**Displaying Perpendicular Lines**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.

**2D Diagram:**

2. Double-click on the curve or its axis label from whose points you would like to drop perpendicular lines onto the X or Y axis.
3. In the dialog box that appears, click on the **Symbols** tab.
4. Check the option **Symbols on**.
5. In the **Type** field, select **Perpendicular on Y axis** or **Perpendicular on X axis**.

**3D Diagram with Space Curve:**

2. Double-click on the curve from whose points you would like to drop perpendicular lines onto a plane.
3. In the dialog box that appears, click on the **Perpendiculars** tab.
4. Select the option that corresponds to the plane onto which you would like to drop the perpendicular line.
5. Set the appearance of the perpendicular lines in the corresponding line attribute fields.

**Editing Perpendicular Lines**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on the perpendicular line that you would like to edit.
3. On the **Drawing Tools** toolbar, select a line color or width.

**Adding a Projection**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve that you would like to project.

**3D Diagram with Space Curve:**

3. In the dialog box that appears, click on the **Projections** tab.
4. Select the planes on which you would like to project the space curve.

5. Set the appearance of the projection lines in the associated line attribute fields.

### **3D Diagram with Surface:**

3. In the dialog box that appears, click on the **Fill** tab.
4. In the **Projection on** field, select the planes on which you would like to project the surface.

### **3D Diagram with Grid:**

3. In the dialog box that appears, click on the **Grid Lines** tab.
4. In the **Projection on** field, select the planes on which you would like to project the grid.

### **Changing the Symbol Size**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on any symbol on the curve. You can select several curves at the same time by holding down the CTRL key and clicking on them. You can select all curves by pressing the ALT key.
3. Resize the symbol in the **Size** field on the **Position** toolbar.

### **Changing the Symbol Shape**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label.
3. In the Properties dialog box of the curve you can set the symbol type in the **Type** drop down box on the **Symbols** tab. If you choose **Automatic**, FlexPro automatically selects a symbol using the number of the curve or the data series in a 3D curve.

### **Changing the Number of Symbols**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on any error indicator of the curve to open its Properties dialog box.
3. On the **Symbols** tab in the **Start at** field, enter the index of the first data point for which a symbol is to be displayed. In the **Spacing** field, specify at what interval the



each data point is to be displayed.

## **Working with Legends**

### **Adding or Deleting a Legend**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the diagram plane to open the Properties dialog box.
3. Click on the **Legend** tab.
4. Check the option **Use legend**.

### **Positioning the Legend**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Select the legend by clicking on its border. If there is no border, click on the background. The legend will now appear with a hatched border.
3. Choose a **position** from the **Position** toolbar to specify on which side of the diagram the legend is to appear.
4. If necessary, use the mouse to move the legend in order to adjust its position.

### **Adding or Deleting a Curve in a Legend**

#### **To add a curve in a legend:**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve that you would like to add to the legend.
3. In the Properties dialog box of the curve, click on the **Legend** tab.
4. Check the option **Add curve to legend**.

#### **To remove a curve from a legend:**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click in the legend on the name of the curve that you want to remove.

3. Uncheck the option **Add curve to legend**.

### Displaying Data Set Information in the Legend

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click in the legend on the curve text that you want to change.
3. In the **Text** input field, place the cursor at the position where you would like to insert a field.
4. Select the desired item from the **Fields** list box. A field is then inserted into the text. This field is later replaced by the updated data set information.

### Adding a Title to the Legend

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the legend title, edge of the legend or diagram surface, but **not** on a curve in the legend.
3. In the Properties dialog box click on the **Legend** tab, if necessary.
4. Select the option **Title**.
5. Enter the desired text in the **Title** input field.

### Changing the Number of Columns in a Legend

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the legend title, edge of the legend or diagram surface, but **not** on a curve in the legend.
3. In the Properties dialog box click on the **Legend** tab, if applicable.
4. Enter the desired quantity in the **Number of columns** field.
5. You can adjust the arrangement of the entries using the **Arrange entries line by line** option.

**Adding or Deleting a Legend Frame**

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the legend title, edge of the legend or diagram surface, but **not** on a curve in the legend.
3. In the Properties dialog box click on the **Legend** tab.
4. Choose a **Color** for the border or select **No Line** so no border will be displayed.

**6.2 Column and Cell Tables**

You can use these presentation objects to display data in table form. You can use the Table Wizard to create column and cell tables, which offers an assortment of templates.

**Column Table**

The column table consists of one or more columns. One data set can be assigned to each column, supplying all of the data for this column. Therefore, the column table is particularly suited for displaying data sets containing more than one value, i.e. data series or data matrices. A typical example would be a data series with the local maxima of a signal. Every column has one column title where fields for the name and unit of the data set can be inserted. Due to this structure, the behavior of the column table is particularly dynamic. The number of values supplied by a data set for a column may vary. The number of rows in the table then changes automatically.

The following illustration shows a column table with an index column, a column with a data series and a column with a data matrix. A column index is also added to the last one.

Column Table

Index Column	Column With Data Series	Column With Data Matrix		
		1	2	3
1	17.11	0	15.95	5.01
2	2.49	3.66	19.07	14.64
3	6.15	10.71	16.98	15.76
4	11.47	18.04	17.81	3.42
5	9.46	15.6	12.57	14.28
6	15.24	19.3	3.71	9.62
7	3.84	3.95	12.34	1.63
8	14.56	18.03	5.3	16.13

**Data Columns, Index Column and Column Index**

Each column in the column table can either display a scalar value, a data series or a data matrix. Aggregate data structures are displayed in several columns. For example, if you add a signal, two columns are added to the column table, one column with the X component of the signal and one column with the Y component of the signal. All data types are permitted, particularly including complex numbers and strings.

If a column represents a data matrix, then this column is divided into several sub-columns, one for each data series in the data matrix. This type of column can have a column index, which is then displayed under the column title.

In addition to the data columns, an index column can be added to the column table, and you can use this column to number the rows in the table. You can give the complete table a text label that you can position above or below it.

**Unit**

You can specify an output unit for each data column or index column. The corresponding data is then converted to this unit before output. If you do not specify an output unit, the data is output in the unit in which it occurs.

**Multi-Page Column Tables**

If you display column tables in the object window or insert them into FlexPro documents, these column tables can be wrapped across several pages in the document. The page wrap occurs automatically if there is not enough space available for a column. FlexPro

first tries to wrap the column on the same page, and the number of columns in the table is multiplied. The column table is not continued on the next page until all of the space on the first page is fully occupied. The titles appear on every page.

---

**Note:** This wrapping option is only available for displaying in the object window and in FlexPro documents. If you export the column table as an OLE object, only the first page is displayed.

---

## Cell Table

The cell table is made up of several cells arranged as columns and rows. Each cell may contain an individual text where you can embed fields for calculation results. The cell table is ideally suited for displaying data sets containing scalar values such as the mean value of a signal. The cell table corresponds to the tables you might already be familiar with from your word processor.

The following illustration shows a typical cell table:

	Quantity	Value
Statistics	Maximum	2.50 V
	Minimum	0.50 V
Mean Values	Arithmetic Mean	1.49 V
	Square Mean (RMS)	1.59 V

## Working with Tables

### Creating a Column or Cell Table

1. In the Folders window, select the folder where you want to place the new column or cell table.
2. Select the data you want to display as columns or cells in the table. You can select data sets, formulas, analysis objects or a cell range in a folder's data view or in the window of an open data set. You can also use the range between the cursors of a diagram as the data selection.

3. Click on **Insert Table** in the **FlexPro Objects** toolbar.
4. The Table Wizard appears to guide you through the remaining steps.

### **Adding or Deleting Dividing Lines and Frames**

#### **To delete a dividing line or the frame, do the following:**

1. Click on the line or frame while holding down the SHIFT key to select the item.
2. On the **Drawing Tools** toolbar, select **No Line** from the **Line Color** menu.

#### **To add a dividing line or the frame, do the following:**

1. Make sure that no item in the table is selected by clicking on a point outside the table.
2. Now use your right mouse button to click on the table and select **Properties**.
3. In the dialog box, select the **Lines** tab.
4. Select the option corresponding to the item to be displayed.

### **Changing the Color of Dividing Lines, Frames, Text and the Background**

1. Select the objects that have a color you would like to change. You can select several objects by holding down the CTRL key and clicking on them. To select the frame or a dividing line, you have to hold down the ALT key while clicking on it.
2. You can change the font color of the selected items by choosing a color in the **Font Color** menu on the **Format** toolbar. Correspondingly, you can change the color of lines and fills using the **Line Color** or **Fill Color** menus on the **Drawing Tools** toolbar.

### **Changing the Text Font**

1. Select all of the text in the table that has a font you would like to change. Press the CTRL key to select several lines of text. Press the ALT key to select all text.
2. Select the desired font from the drop-down list of the same name in the **Format** toolbar.

**Changing Text Alignment for the Column Title or Cell Text**

1. Click on the column title or cell text that has the alignment you would like to change.
2. On the **Format** toolbar, click on **Left-justified**, **Centered** or **Right-justified** to change the alignment.
3. The text in a table cell can be rotated 90° so that it appears vertical. To do this, select **90°** in the **Angle** field on the **Position** toolbar.

**Adding or Deleting a Label****To delete a label, do the following:**

1. Double-click on the label.
2. In the dialog box that appears, uncheck the option **Labeling on**.
3. If the label is not completely displayed, you may have to increase the height of the table by dragging the handle on the edge down or by using the command **Adjust Size To Content** from the **Table** menu.

**To add a label, do the following:**

1. Make sure that no item in the table is selected by clicking on a point outside the table.
2. Now use your right mouse button to click on the column table and select **Properties**.
3. In the dialog box, select the **Labeling** tab.
4. Check the option **Labeling on**.
5. Enter a text for the label.

**Positioning the Label**

1. Click on the label in the table to select it.
2. On the **Position** toolbar, open the **Position** menu and select the desired position.

**Embedding Graphics**

1. Choose the command **Insert Image** from the **Table** menu.
2. Choose a file in the **Insert Image** dialog box.
3. Click OK to close the dialog box. The newly inserted image is now displayed.
4. If necessary, drag the image with your mouse to position it.

5. You can change the absolute height and width as well as the relative size in relation to the original image size in FlexPro's Properties window.

## Working with Column Tables

### Adding a Data Column

The easiest way to add a data column to a column table is to use the mouse to drag the data set to be displayed in the column from the **Object List** into the column table. The new column is then inserted at the location selected and adopts the style attributes of its left neighbor.

However, you can also insert an empty column and then manually enter the data set:

1. Double-click on the column table in the worksheet or document to open it. Now use your right mouse button to click on the column table.
2. In the **Table** menu, select **Insert Data Column**.
3. In the dialog box that appears, enter the data set for the column. If the data set has an aggregate data structure, you can select a component, such as `Signal.X` for the X component of a signal.
4. Click on **OK** to add the column.
5. If the new column is not displayed completely, you may have to increase the width of the table by dragging the handle on the edge to the right or using the command **Adjust Size To Content** from the **Table** menu.

### Adding an Index Column

1. Double-click on the column table in the worksheet or document to open it. Now use your right mouse button to click on the column table.
2. In the **Table** menu, select **Insert Index Column**.
3. If the new column is not displayed completely, you may have to increase the width of the table by dragging the handle on the edge to the right or using the command **Adjust Size To Content** from the **Table** menu.

### Specifying the Output Unit

1. Double-click on the data of the column table or column index for which you want to specify an output unit.
2. In the Properties dialog box that appears, enter the preferred output unit in the **Unit** field or choose the unit from the list.



3. Close the dialog box by clicking **OK**.

---

**Note:** If you specify an output unit, the data must be present in a compatible unit.

---

### Rearranging Columns

To move a column in a column table to a different location, use your mouse to drag the column to the desired location and drop it there.

---

**Note:** You can also rearrange the columns on the Columns tab of the Properties dialog box.

---

### Changing the Column Width

#### To change the width of a column:

1. You may first have to increase the width of the table by using your mouse to drag the handle on the edge to the right.
2. To change the width of a column, use your mouse to drag and move the right edge of the column.

#### To adjust the width of all columns:

- Use the command **Space Columns Evenly** from the **Selection** menu.

---

**Note:** The column width that you set is a minimum width. FlexPro automatically increases the column width if the data cannot be displayed using the width specified.

---

### Changing the Height of the Rows and Column Titles

1. You may first have to increase the height of the table by dragging the handle on the edge downward with your mouse.
2. To change the height of a row, use your mouse to drag and move the bottom edge of the row.

**Note:** You can set the row height for the title and the data rows separately; however, the setting for rows with data applies for all data rows. The row height is a minimum height. FlexPro automatically increases the size of the row if the content cannot be displayed with the height specified.

---

### Navigating Multi-Page Column Tables

- Use the commands **Previous Page** or **Next Page** in the **View** menu or in the **Drawing Tools** toolbar to go to the previous or next page of the table.
- To show the first or last page of the table, use the commands **First Page** or **Last Page** in the **View** menu.
- You can use the command **Choose Page** in the **View** menu to select any page.
- You can use the mouse wheel to scroll through several pages of the table.

### Enabling/Disabling the List View with Alternating Colors

#### To turn off the list style, do the following:

1. Double-click on a row that is filled with the alternate color.
2. In the dialog box that appears, uncheck the option **Color every second row with the following color**.

#### To turn on the list style, do the following:

1. Make sure that no column in the column table is selected by clicking on a point outside the table.
2. Now right-click with your mouse on the table and select **Properties**.
3. In the dialog box, select the **Columns** tab.
4. Select the option **Color every second row with the following color**.
5. Click on the color field to set the color.

### Changing the Data Format in a Column Table

#### To change the formatting of all values of a column, do the following:

1. Double-click on any value in the column with the format that you would like to

change.

2. Click on the button in the **Format** field to adjust the formatter.
3. Select the desired format.

**Using the Properties window, you can set formatting for several columns at once:**

1. If the **Properties** window is not already open, click on **View** in the menu bar, point to **Window** and then click on **Properties**.
2. Select the columns that have formatting that you would like to change by clicking on these while holding down the CTRL key. You can select all columns by pressing the ALT key.
3. Now click in the **Format** field in the Properties window and then click on the button that appears to the right of the box.
4. Select the desired format.

---

**Note:** The format applies to all values in the column(s) selected.

---

**Displaying the Unit Symbol after the Data Values**

1. Double-click on the data of the column table or column index for which you want to specify an output unit.
2. In the Properties dialog box that appears, click on the button in the **Format** field.
3. In the **Formatter Properties** dialog box and on the **Numeric Value** tab, select the option **Append unit**.
4. Close both dialog boxes by clicking **OK**.

---

**Note:** If you specify an output unit, the data must be present in a compatible unit.

---

**Adding or Deleting a Column Title**

**To delete a column title, do the following:**

1. Double-click on the column title that you would like to delete.
2. In the dialog box that appears, uncheck the option **Column title on**.

**To add a column title, do the following:**

1. Double-click on the column to be given a title.
2. In the dialog box, select the **Column Title** tab.
3. Check the option **Column title on**.
4. Enter text for the column title. You can embed fields for displaying header information from the data set.

**Displaying Data Set Information in the Column Title**

1. Double-click on the column with the title into which you would like to insert header information from the data set.
2. In the dialog box, select the **Column Title** tab.
3. Check whether the option **Column Title** is selected. If not, then select this option now.
4. In the **Column Title** field, place the cursor at the position where you would like to insert a field.
5. Choose the desired item from the **Fields** list box. A field is then inserted into the text. This field is later replaced by the updated data set information.

**Adding or Deleting a Column Index**

**To delete a column index, do the following:**

1. Double-click on the column index that you would like to delete.
2. In the dialog box that appears, uncheck the option **Column index on**.

**To add a column index, do the following:**

1. Double-click on the column to be given an index.
2. In the dialog box, select the **Column Index** tab.
3. Check the option **Column index on**.
4. Select **Data Set** and specify a data set if you do not want to number the columns, but want to display the data from a data set instead. This option can be used, for instance, to display the Z component in a Signal Series.

**Note:** A column index can only be used for columns with a data set that supplies a data matrix.

---

## Working with Cell Tables

### Adding a Column

1. Double-click on the cell table in the worksheet or document to open it.
2. Click to select any cell in the column before which you want to insert a new column. If you want to add a new column behind the last column, no cell in the cell table should be highlighted.
3. Now use your right mouse button to click on the table and select **Insert Column**.
4. If the new column is not displayed completely, you may have to increase the width of the table by dragging the handle on the edge to the right or using the command **Adjust Size To Content** from the **Table** menu.

### Adding a Row

1. Double-click on the cell table in the worksheet or document to open it.
2. Click to select any cell in the row before which you want to insert a new row. If you want to add a new row behind the last row, no cell in the cell table should be highlighted.
3. Now right-click with your mouse on the table and select **Insert Row**.
4. If the new row is not displayed completely, you may have to increase the height of the table by dragging the handle on the edge down or by using the command **Adjust Size To Content** from the **Table** menu.

---

**Note:** If you have highlighted the last cell in the bottom right-hand corner, you can easily add a new row by pressing the TAB key. The same thing works if you select the first cell in the upper left-hand corner and press the TAB key while holding down the SHIFT key.

---

### Deleting Columns or Rows

1. Select the column or row that you want to delete by clicking on an area below or to the right of the edge of the column or row. You can select several columns or rows by holding down the CTRL key.
2. Use the command **Delete Rows or Columns** from the **Selection** menu.

### Rearranging Columns or Rows

1. Select the column or row that you want to move by clicking on an area below or to the right of the edge of the column or row.
2. Use your mouse to drag the selected column or row to the desired location and drop it there.

### Specifying the Output Unit

1. Double-click on a value displayed in a table cell in order to change its output unit.
2. In the dialog box that appears, place the insertion point on the field to be changed, directly after the { character, which indicates the field. A typical field, for instance, is %&{Velocity}.
3. Enter the operator Unit<Unit Symbol> directly after the { character, e.g. %&{Unit<km/h> Velocity}.
4. Close the dialog box by clicking **OK**.

---

**Note:** If you specify an output unit, the data must be present in a compatible unit.

---

### Changing the Column Width

#### To change the width of a column:

- Move the right margin of the column by dragging it with the mouse.

#### To adjust the width of several columns:

1. Select the columns you want to make the same width by clicking on the area below the edge of the columns while holding down the CTRL key. You can select all columns by pressing the ALT key.
2. Click on the **Selection** menu and then on **Space Columns Evenly**.

**Note:** The column width that you set is a minimum width. FlexPro expands the column automatically if the text cannot be displayed using the pre-configured width.

---

## Changing the Row Height

**To change the height of a row, do the following:**

- Move the lower edge of a row by dragging it with your mouse.

**To adjust the height of several rows, do the following:**

1. Select the rows that you want to adjust for height by clicking on a location to the right of the row while holding down the CTRL key. You can select all rows by pressing the ALT key.
2. Click on the **Selection** menu and then on **Space Rows Evenly**.

---

**Note:** The row height is a minimum height. FlexPro expands the row automatically if the text cannot be displayed properly using the given height.

---

## Editing Cell Text

1. Double-click on the cell table in the worksheet or document to open it.
2. Select the cell with text that you want to edit or in which you want to enter new text.
3. To insert new text into the cell, simply type it in.
4. If you want to edit existing text, press F2 or click again on the cell that is already selected and then edit the text.
5. If you not only want to enter text, but also want to format it and insert fields, press the ENTER key or double-click on the cell. A dialog box appears, allowing you to edit the text.

## Displaying Data in a Cell

1. Use your mouse to drag the data set from the Object List onto any cell in the cell table.
2. In the cell text, a field is automatically inserted for the value and the unit of the data set.

### Changing the Display Format of Data in a Cell

1. Double-click on the value in a cell of the table with values for which you would like to change the formatting.
2. Place the cursor onto the formatter to be changed, directly after the % character, which indicates the formatter. A typical formatter, for instance, is %&{DataSet}.
3. Click on **Edit Formatter** above the input field to set the formatter.
4. Select the desired format.

### Merge Cells

1. Double-click on the cell table in the worksheet or document to open it.
2. Select the adjacent cells that you want to merge into one cell by clicking on them while holding down the CTRL key.
3. Use your right mouse button to click on the table and then click on **Merge Cells**.

---

**Note:** When merging cells, the cell in the upper left-hand corner of the selected range is enlarged so that it covers all of the other cells. You can use the command **Split Cells** to undo this step, making the hidden cells visible again.

---

## 6.3 Text

This presentation object shows a block of text with embedded calculation results. You can freely format the text as you would in a word processing program. To display values and other information, insert fields into the text. These fields are later replaced by data from data sets or by other information.

### Working with Text

#### Creating Text

In the Folders window, select the folder where you want to place the new text.

1. Use your right mouse button to click on the Object List, which will open the context menu.
2. Select the **New** menu and then **Text**.



### Editing Text

Before you can edit a text object, you have to open it. To do this, double-click on the text in the Object List. A window appears where you can then edit the text.

The text window has two views. In **Draft** view, enter the text and insert fields for calculated results. In **Preview**, the text is displayed using the replaced fields.

### Deleting Text

1. Select the text that you would like to delete.
2. To remove the text for re-insertion into a different location in the document, click on **Edit** and then **Cut**, or hold down the DELETE key to delete the text without copying it to the clipboard.

---

**Note:** Click anywhere in the text to cancel a selection. To undo a deletion, click on **Edit** and then on **Undo**.

---

### Copying Text

1. In the text containing the information to be copied, while holding down the left mouse button, drag the cursor across the information you would like to select.
2. In the **Edit** menu, click on **Copy**.
3. In the text where the information is to be inserted, click on the location where you would like to place the information.
4. In the **Edit** menu, click on **Paste**.

---

**Note:** You can insert the information multiple times.

---

### Searching for Text

1. Click on the position in the text where you would like to start the search.
2. In the **Edit** menu, click on **Find** and enter the text you want to search for in the **Find what** field.
3. To find another location with the same text, continue by clicking on **Find Next**.

---

**Note:** To find another location with the same text after you have closed the **Find** dialog box, press the F3 key.

---

To search and replace text, click under the **Edit** menu on **Replace** instead of **Find**.

---

### Searching and Replacing Text

1. Click on the location in the text from which you would like to start replacing text.
2. In the **Edit** menu, click on **Replace** and then enter the text you want to find as well as the text that will replace it.
3. To replace the found text throughout the document without confirmation, click on **Replace All**.

To replace the found text at the particular places where it appears, click on **Find Next** and then **Replace**.

### Changing the Font, Font Style and Font Size

1. Select the text that you would like to format.
2. In the **Selection** menu, click on **Font**.
3. Click on the desired options.

---

**Note:** You can specify the font for new text by changing the settings for the font before you start to enter the text.

You can more easily format the text by using the **Format** toolbar.

---

### Formatting Paragraphs

1. Click on the paragraph that you would like to format.
2. In the **Selection** menu, click on **Paragraph** and select the desired justification and indents.

---

**Note:** To set the paragraph format for new text, change the format settings before you start to enter the text.

---

### Setting Tabs in Paragraphs

1. Select the paragraph for which you would like to set tabs.
2. In the **Selection** menu, click on **Tabs**.
3. To set a tab, enter the position in the **Tab position** field. Next, click on **Set**.

**Note:**

- To delete a tab, open the **Tabs** dialog box from the **Selection** menu, select it in the list of tabs and then click on **Clear**.
  - To delete all of the tabs in the paragraph selected, click on **Clear All** in the **Tabs** dialog box.
  - To place tabs with the ruler, click on the ruler where the tab stops are to be set.
  - To delete tabs with the ruler, use your mouse to select them and drag them from the ruler.
- 

**Creating a List**

1. Place the cursor at the location in the document where the list is to start.
2. Click on **Bullets** in the **Selection** menu. Now enter the text for the list.
3. If you press the ENTER key, another bullet appears in the next line.
4. To end the list, click on **Bullets** again in the **Selection** menu.

**Inserting a Date or Time Field**

1. Click on the location in the text where the field is to be inserted.
2. On the **Text Fields** toolbar, click on **Insert Date** or **Insert Time** to add the current date or time.

---

**Note:** The field is replaced by the corresponding part of the system date whenever the text is updated.

---

**Inserting a Field to Display Data or Other Elements**

1. Click on the location in the text where the field is to be inserted.
2. On the **Text Fields** toolbar, click on **Insert Field**. The Field Wizard appears to guide you through the remaining steps.

**Inserting Fields for the Data Set Name, Value and Unit**

You can display the data set name, value and unit in a text object by simply using your mouse to drag the data set from the Object List into the text object and drop it at the location where the fields are to be inserted.

### Inserting a User Prompt

1. Click on the location in the text where the field is to be inserted.
2. On the **Text Fields** toolbar, click on **Insert Field** to activate the Field Wizard.
3. Select the **Enter** category and the **User Input** field.
4. Click on **Next** and follow the Wizard's instructions.

### Post-Editing a Field Formatter

1. Place the cursor on the formatter to be changed, directly after the % sign, with which the formatter starts. A typical formatter is, for instance, %f{DataSet}.
2. Now click on **Edit Formatter** in the **Text Fields** toolbar. A dialog box appears where you can set the desired options.

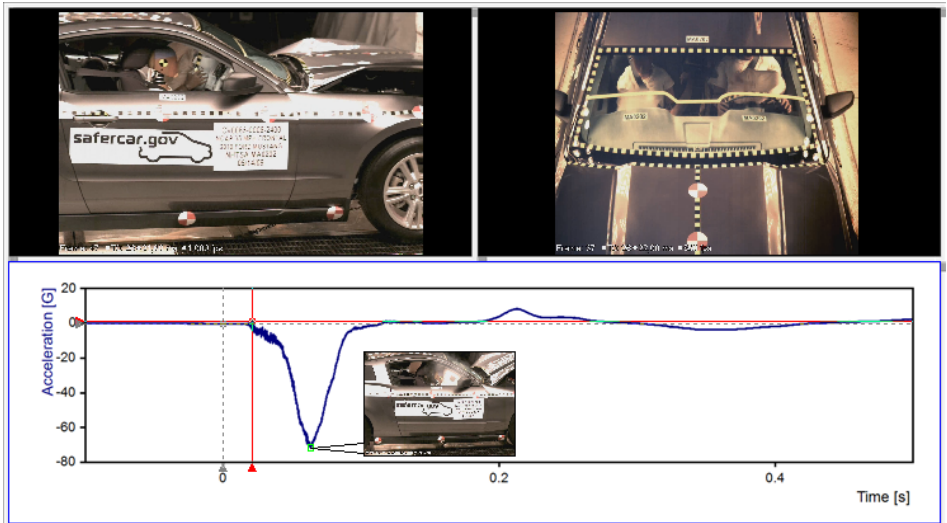
## 6.4 Media

You can use the **Media** presentation object to play video and audio files and to analyze videos and audio signals that you have recorded synchronously with physical quantities.

---

**Note:** This presentation object is only available in FlexPro Professional and FlexPro Developer Suite.

---



You can use media such as diagrams and tables and embed or link them in documents and worksheets.

## Time Base

The frame number or time assigned to the frame is used to assign frames in a video. The time base assigns these times to the frames. FlexPro supports three modes:

- **From media file**

The time information is determined from the number of frames and the playback length of the video. The time 0 s is assigned to the first frame in the video.

- **Starting time and frame rate**

In this case, enter the start time and the frame rate. The start time is the time in seconds assigned to the first frame in the video. The frame rate specifies the number of frames per second. You can use this time base for videos whose frame rate specified in the media file deviates from the frame rate in which the video was originally recorded. This is usually the case for videos recorded on high-speed cameras at 1000 frames per second but are to be played back at 24 frames per second, for instance.

- **Two frames with frame times**

The time information is calculated from the times of two random frames. These variants are ideal for videos with opening credits, for instance, in which the time of the first frame is not really known. You can enter the times either using the format +-HH:

MM:SS,sss or using any time unit, such as **-25.3 ms**. For the frame numbers, enter the one-based numbers of the corresponding frames.

- **From data set**

Here you enter a data series with the time values for all frames. The time values must be increasing, but not equidistant. Use this setting for videos recorded at variable frame rates using an external timer.

---

**Note:** Some video formats do not support positioning via frame numbers. The same naturally also applies to all audio formats. In this case, FlexPro adopts a total of 24 frames or positions per second to which you can move. The number of frames or navigation positions are then the result of the playback length of the media file.

---

### Still Image

The Media object takes an individual image from the video and displays it when the video is not playing. Likewise, when you print out a document in which you placed a Media object, this still image is printed. You can choose the still image using the cursor or by entering the time of the image you want displayed in the Properties dialog box. As in other presentation objects, you can draw in the Media object in order to mark points of interest in the still image with arrows.

### Compatible Audio and Video Formats

The Media object is based on the same technology as Windows Media Player. All formats that Windows Media Player can play are therefore also compatible with FlexPro. If a media file cannot be played, you can subsequently install the relevant Windows Media Player codec.

## Working with Media

### Creating Media

#### Adding Media Manually

1. In the Folders window, select the folder where the new diagram is to be placed.
2. If you want to embed the new media into a worksheet or document at this time, click on the open worksheet or document window. Otherwise, click in the Object List.
3. Click **Insert Media** from the **FlexPro Objects** toolbar.
4. In the dialog box that now appears, enter the path name of the media file on the **Source** tab.

**Note:** The media file is not copied to the object, but has to be found under the specified path name each time it is played. Therefore, you should either save your media files in a folder of your choosing or copy them to the same folder in which your FlexPro project database resides. If you choose the second method, you should only enter the file name and not the full path name in the Media object so that the link remains when you copy your project.

---

### **Adding Media via Drag-and-Drop**

- Select the media file in Windows Explorer and use drag-and-drop to drag it to the FlexPro Object List or to an open worksheet or document.

### **Creating the Time Base**

1. Right-click with your mouse on the media and select **Properties**.
2. In the Properties dialog box that now appears, click on the tab called **Time Base**.
3. Choose the preferred **Time Base** and adjust the necessary settings. Launch Help in the dialog box if necessary.

### **Playing Media**

#### **Playing Media from the Object List**

1. Double-click on the object to open it.
2. Click **Playback** on the FlexPro **Media** toolbar.
3. Click **Pause** to pause playback.
4. Click **Stop** to stop playback.

---

**Note:** The media is played back from the beginning. If you want to play the media back from the position of the still image displayed, you have to activate the cursors before starting playback. When you disable the cursors again, FlexPro uses the current position as the new still image. If you want to keep the original still image, you have to choose **Restore Original Image Section** from the **Cursors** menu before deactivating the cursors.

---

### Playing Media from a Document or Worksheet

1. If the media is in a document, click **Toggle Cursors** on the **Cursors** toolbar.
2. Click on the media that you want to play.
3. Click **Playback** on the FlexPro **Media** toolbar.
4. Click **Pause** to pause playback.
5. Click **Stop** to stop playback.
6. If necessary, click **Restore Original Image Section** on the **Cursors** toolbar to restore the originally displayed still image.

### Adjusting the Volume

#### With Cursors Enabled

- Move the vertical scrollbar on the media object to adjust the volume.

#### With Cursors Disabled

- Enter the volume as a percentage in the Properties window.

### Changing Playback Speed

- Enter the ratio between the desired and "natural" playback speed in the **Playback Speed** box on the **Media** toolbar.

---

**Note:** Not all media formats support the ability to increase or reduce playback speed. In this case, the speed set for the next time the media is played is reset to 1:1.

---

### Choosing the Still Image for Display

#### With Cursors Enabled

- In the Properties window enter the zero-based position index of the preferred still image or its X position in seconds.

**or**

- Use the horizontal scrollbar to navigate to the desired image.



### **With Cursors Disabled**

- In the Properties window, enter the one-based frame number of the preferred still image or its position either in the format +-HH:MM:SS,ssss or in any time unit, such as "120.3 ms".

## **6.5 Documents**

FlexPro lets you create and manage multi-page documents directly in the project database. In this type of document you can then embed diagrams, text and tables or create links to objects within the project database.

The powerful FlexPro Document Editor also lets you create your own drawings, such as the sketch of a test setup, thereby creating expressive and appealing documents. You can also embed any OLE objects and images, such as your company logo, into the documents.

### **Headers and Footers**

You can create headers and footers for individual drawing objects by enabling **Show On All Pages**. The drawing objects will then appear in the same location on every page of the document.

### **Multi-Page Presentation Objects**

Using the internal FlexPro Document Editor offers you additional options that your word processing program does not have. For example, 2D diagrams and column tables can be wrapped to the following pages automatically. Here, the number of pages is adjusted automatically to the volume of data.

### **Page Layout**

The document page layout specifies the size of the pages and margins in the document. The page size does not necessarily have to correspond to the size used for output to a printer. FlexPro can adjust the document to the paper format later. However, if you prefer to print to scale, you should use the page size of your printer for your documents.

The page margins are only there to provide onscreen visual orientation. FlexPro lets you draw outside the margins that have been set.

### **Drawing Object Alignment Options**

If you have created a group of drawing objects, you can align the entire group to the

left, right, top or bottom of the page or centered on the page. You can align drawing objects relative to each other or to the margins of pages. You do not have to select or move the objects individually.

You can also align objects using the grid. When you draw or drag objects, the corners are aligned to the nearest grid intersection point. Even if the grid is not visible, it still helps you to align the objects automatically. You can change the grid spacing and permanently or temporarily deactivate automatic alignment to the grid.

### **Grouping Drawing Objects**

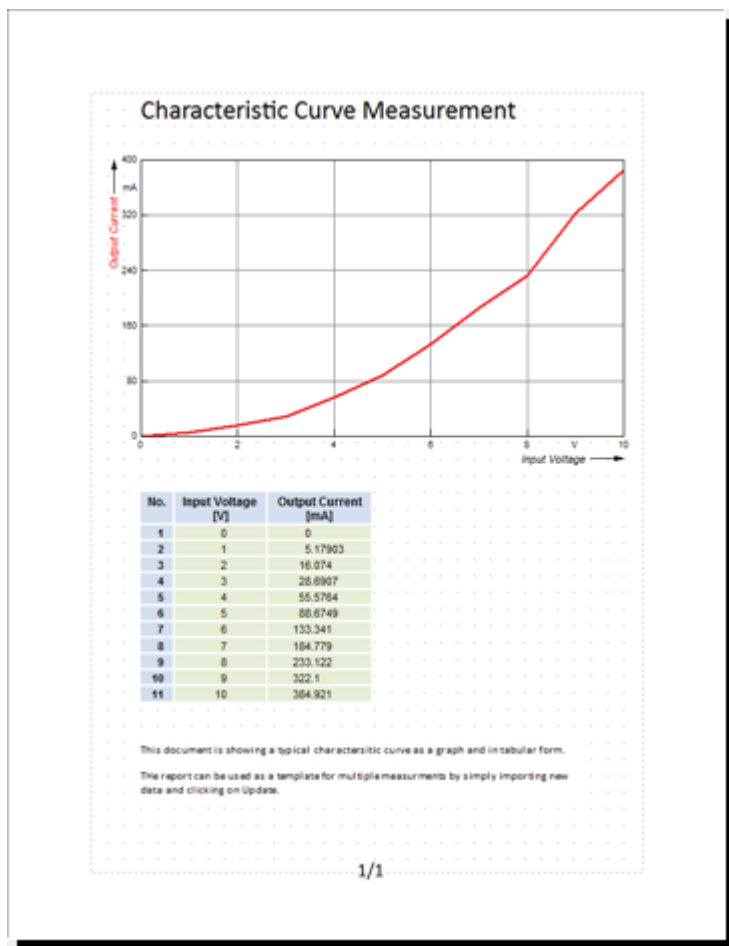
If you combine several drawing objects into a group, you can edit the objects as if they were one object. You can assign a property to all individual objects of a group simultaneously, such as hatching or a line color. The objects can also be resized, moved, tilted and rotated as a group.

Grouping the drawing objects is also very useful when creating complex graphics. You can create groups within groups and thus create a group hierarchy. You can draw a set of objects and group the objects, then draw the next set and group it, etc. Thus, if you need to edit a section of the drawing, you can undo the grouping for this section without affecting the other groupings.

### **Using Cursors in the Document**

You can also measure diagrams using cursors in the document. While the cursors in the worksheet are always active, in the document, you have to activate them first. You then have all Cursor.

The following illustration shows a typical document:



## Working with Documents

### Creating a Document

1. In the Folders window, select the folder where you want the new document to be placed.
2. Right-click with your mouse on the **Object List** to open the context menu.
3. Select the **New** menu and then **Document**.

### Setting Up the Page Format

1. Open the document by double-clicking on its name in the **Object List**.
2. In the **File** menu, click on **Page Setup**.
3. Change the settings as needed.

---

**Note:** The page margins are only there to provide onscreen visual orientation. FlexPro lets you draw outside the margins that have been set.

---

### Adding and Deleting Pages

- Use the **Insert Page After** or **Insert Page Before** command in the **Document** menu to add a blank page after or before the current page. The new page will now appear in the window and can be edited.
- Use the **Delete page** command in the **Document** menu to remove the page currently displayed in the document.

### Inserting Page Numbers

1. Select **Insert Page Number** from the Document menu. The page number is inserted at the bottom of the document.
2. If necessary, drag the page number field with your mouse to position it.
3. Double-click on the page number if you want to edit it.

### Navigating Documents

- Use the commands **Previous Page** or **Next Page** in the **View** menu or in the **Drawing Tools** toolbar to go to the previous or next page of the document.
- To view the first or last page of the document, use the commands **First Page** or **Last Page** in the **View** menu.
- Use the command **Choose Page** from the **View** menu to select any page.
- You can use the mouse wheel to scroll through several pages of the document.

### Resizing Graphics

- From the **Zoom** submenu of the **View** menu, you can select a display size between 25% and 200%. You can also set the display size to 100% by clicking on the **Drawing Tools** toolbar if you click on the symbol **Zoom to 100%**.
- Alternatively, you can change the display size using the mouse by turning the mouse wheel while holding down the CTRL key.
- You can also set the display size is automatically so that the document remains completely visible in the window. To do this, use the command **Fit To Window**. You can use the **Magnifier** to display any section close up. Simply drag a rectangle to enclose the section.
- A grid can be displayed in the background to make aligning objects easier. You can activate or deactivate the grid from the **View** menu. You can set the grid spacing and the width of the margins using the **Page Setup** command under the **File** menu.

### Displaying Multiple Adjacent Pages

1. Choose the **New Window** command from the **Window** menu to open a second window of the same page.
2. In the **Window** menu, click on **New Vertical Tab Group** to display both windows side by side.
3. Now you can navigate both windows independently using the **Next Page** and **Previous Page** icons on the **Drawing Tools** toolbar.

### Changing the Drawing Grid

1. Choose the **Page Setup** command from the **File** menu.
2. Specify the preferred horizontal and vertical grid spacing.

### Activating or Deactivating the Automatic Snap to Grid Feature

Use the **Snap to Grid** option from the **Drawing** menu to activate or deactivate automatic alignment to the grid.

---

**Note:** You can temporarily deactivate the automatic snap to grid feature by holding down the ALT key.

---

## **Linking and Embedding Presentation Objects**

### **Inserting a link from a presentation object created in the project database into the document**

1. Use your mouse to drag the object from the Object List directly to the location in the document where you want it to appear.
2. Position the object and adjust its size.

### **Embedding a new presentation object into the document**

1. In the Object List, select the data sets that you want displayed in the object.
2. Use your right mouse button to click on the document.
3. In the context menu, point to **New**.
4. Select the object type, such as diagram. The Wizard appears to guide you through the rest of the creation process.
5. Position the object and adjust its size.

### **Displaying a Data Set in a New Embedded Diagram**

Drag the data set from the Object List to the position in the document where the diagram is to be placed. Depending on the data structure of the data set, there are various display options from which you can choose. A new embedded diagram is created in which the data set is displayed as a curve.

### **Embedding a copy of an existing presentation object into the document**

1. While holding down the CTRL button, use your mouse to drag the object from the Object List or from another document directly to the location in the document where you want it to appear.
2. Position the object and adjust its size.

### **Move an existing presentation object into the document**

1. While holding down the ALT button, use your mouse to drag the object from the Object List or from another document directly to the location in the document where you want it to appear.
2. Position the object and adjust its size.

**Note:** If you want to copy or move an object from a worksheet, you have to use the clipboard.

---

### Editing Presentation Object Links

1. Select the object link in the document that you want to edit.
2. Select **Edit** from the **Selection** menu.
3. Select any presentation object from the **Choose Object** dialog box to which you want the link to refer.

### Converting a Presentation Object into a Drawing

1. Select the presentation object in the document that you want to convert into a drawing.
2. Choose **Convert into Drawing** from the **Selection** menu.

---

#### Note:

- Text objects cannot be converted into drawings.
  - During conversion, the object is replaced by a representation of it, which means the properties of the converted objects can no longer be changed, cursors can no longer be used and the object can no longer be updated.
  - The drawing objects of the converted object are grouped in multiple layers. You may have to ungroup them before you can edit individual drawing objects.
- 

### Embedding Graphics

1. Choose the command **Insert Image** from the **Document** menu.
2. Select a file in the **Insert Image** dialog box.
3. Click OK to close the dialog box. The newly inserted image is now displayed.
4. If necessary, drag the image with your mouse to position it.
5. You can change the absolute height and width as well as the relative size in relation to the original image size in FlexPro's Properties window.

### Linking and Embedding OLE Objects

FlexPro can embed any graphics, documents, tables, etc., which originate from other applications, into documents. The application from which an object is inserted must have an OLE server interface. These objects are called OLE objects.

1. Choose the command **Insert New Object** from the **Edit** menu.
2. Carry out one of the following actions:

#### Inserting an Existing File as an Object

1. Select **Create from File** if the object already exists as a file.
2. Enter the file name and path on the hard drive or click on **Browse** to browse for the file.
3. If you want to insert a link to the file, select **Link**. In this case, you can edit the linked object later using the OLE server application associated with that file type.

#### Inserting a New Object

1. Select **Create New**.
  2. Select an object type.
3. Click on OK.

### Editing an Embedded or Linked OLE Object

1. Select the object that you want to edit.
2. In the **Edit** menu, select the submenu **Object** and then either **Open** or **Edit**. If you select **Open** or if the OLE object is a linked object, the object will open in a separate window. Otherwise, you can edit the object directly in the FlexPro document. The toolbars will then appear for the OLE server application assigned to the object.
3. You can end the process by using the ESC key or by clicking outside of the object window.

### Editing OLE Object Links

- In the **Edit** menu select **Links**.

---

**Note:** In the **Links** dialog box, you can update individual links or switch to automatic or manual update. With **Change Source**, you can refer a different file for the object link.

---



## Working with Drawings and Other Objects

### Repeating Text and Graphics on Every Page

1. Select all of the objects to be displayed on all pages of the document and not just on the current page. You can click on the objects while holding down the CTRL key, or you can use your mouse to drag a rectangle to select an area.
2. From the **Selection** menu, choose the submenu **Show** and then **On All Pages**.

### Rearranging Drawings and Other Objects on a Plane

When several drawing objects are superimposed over each other, you can bring objects to the front or send them to the back.

1. Select all of the objects whose arrangement you would like to change. You can click on the objects while holding down the CTRL key, or you can use your mouse to drag a rectangle to select an area.

#### To bring selected objects one step closer to the foreground or one step further to the background:

2. On the **Drawing Tools** toolbar, click on **Bring Forward** or on **Send Backward**.

#### To bring selected objects closer to the foreground or send them further to the background:

2. On the **Drawing Tools** toolbar, click on **Move to the Foreground** or on **Move to the Background**.

---

**Note:** Objects that are displayed on all pages always appear behind those that are only displayed on the current page.

---

### Aligning Drawing Objects

You can align drawing objects relative to each other or to the page:

1. Hold down the CTRL key and click on the objects you want to select for alignment.
2. On the **Drawing Tools** toolbar, click on **Align Selection**.
3. Select the desired options.

---

**Note:** If you align an object related to the page and later change the page format, you will have to re-align the object to account for the new page dimensions.

---

### Rotating Drawing Objects

1. Hold down the CTRL key and click on the objects you want to rotate.
2. On the **Drawing Tools** toolbar, click on **Rotate 90°** to rotate the objects 90° counter-clockwise.

---

**Note:** Presentation objects and OLE objects cannot be rotated. A presentation object can, however, be converted into a drawing, which can then be rotated.

---

### Grouping Drawing Objects

1. Select all of the objects that you want to group together. You can click on the objects while holding down the CTRL key, or you can use your mouse to drag a rectangle to select an area.
2. Click on **Group** in the **Drawing Tools** toolbar.

### Ungrouping Drawing Objects

1. Click on the group containing the objects that you want to separate.
2. Click on **Ungroup** from the **Drawing Tools** toolbar.

## 6.6 Drawing Objects

You can make the optical design of your FlexPro diagram and table documents and the presentation objects more interesting by using drawing objects. Use the buttons on the **Drawing Tools** toolbar to create lines, arrows, ellipses, rectangles, circles, arcs, polylines, polygons and labels. After drawing an object, you can give it a color or a pattern, change the line type and color of the object, or change, move, rotate or resize it. In the document you can also combine the objects drawn with graphics, which you have inserted into the document. You can copy objects and insert the objects into a different location in the document or into a different document or presentation object.

You can re-shape polylines and polygons at a later time. You can combine any freeform surfaces from arcs, lines and polylines and then fill them or give them a pattern.

Use the **Label** drawing object for smaller amounts of text that you want to insert into a document or a diagram. Larger blocks of text that use different fonts and font sizes should be embedded into the document as text objects.

## Working with Drawing Objects

### Drawing a Line, Rectangle, Ellipse, Arc or Other Drawing Objects

1. Click on the desired button on the **Drawing Tools** toolbar. For example, click on **Line** to draw a line, polyline or polygon.
2. Click on the location where you want to start the line.
3. Now move your mouse to the location where you want the line to end or where you want a corner point of a polyline or polygon.
4. Double-click to set the end point. To set the corner point, click and proceed with the second step. You can obtain a polygon if you place the end of the last line onto the beginning of the first one.

---

#### Note:

- You can influence the appearance of the various drawing objects if you keep the SHIFT key and/or the CTRL key pressed while you draw. By doing this, you can, for instance, change the direction of an arc, a circular arc or a square, or force horizontal or vertical lines to be created.
  - You can cancel drawing an object by pressing ESC.
- 

### Adding Labels

1. Click on the **Label** button on the **Drawing Tools** toolbar.
2. Click on the location where you want to start the label.
3. In the dialog box that appears, enter the text for the label and choose the font.
4. In the text you can embed fields for the page number, data, etc. by placing the insertion point at the desired location and then selecting a field from the **Fields** list box.
5. You can also add a field to display calculation results by clicking on **Insert Field**.
6. You can enter any angle between 0 and 360° in the **Writing direction** field.

---

**Note:** Only use the labeling tool for smaller labels, not for longer texts. If you require more freeform text design, you should use the text object. Labels cannot be as freely formatted as text. However, unlike the text object, you can change the writing direction.

---

### Cutting Drawing Objects

You can cut out drawing objects to combine the fragments into freeform surfaces later, for instance.

1. Click on the **Cut** button on the **Drawing Tools** toolbar.
2. Click on the location where you want to start the cutting edge.
3. Move the mouse to the position where you want to end of the cutting edge.
4. Click to cut.

---

**Note:** Use the **Cut** tool for actions such as cutting out any arc from a circle. The arc tool can only be used to draw 90° arcs.

---

### Connecting Drawing Objects to Freeform Surfaces

You can combine any freeform surfaces, e.g. rectangles with rounded corners or sections of circles from polylines, lines and arcs. You can then give these surfaces a fill pattern, for example.

1. Arrange several polylines, lines and arcs so that they form a closed border. Make sure that the ends are as flush as possible next to one other.
2. Select all of the objects that are to form the freeform surface.
3. Choose the command **Connect** from the **Selection** menu. The selected objects are transformed into a closed polygon to which you can add a fill pattern, for instance.

### Adding and Deleting Arrowheads

1. Click on the arc or the line to which you would like to add one or two arrowheads.
2. Click on **Arrow type** in the **Drawing Tools** toolbar and select an arrow type.

**Adding, Changing or Deleting a Fill Color or a Pattern**

1. Select the objects with the fill color or pattern you would like to change. You can select multiple objects by pressing the CTRL key.
2. On the **Drawing Tools** toolbar, click on **Fill Style** and select the desired pattern.
3. Click on **Fill Color** on the **Drawing Tools** toolbar to select a color. Click on **No Fill** to remove the fill.

**Reusing Display Attributes****To reuse an existing object's display attributes for new objects:**

1. Make sure that no objects are selected by clicking on an empty area in the document or presentation object.
2. Click on **Adopt Attributes** on the **Drawing Tools** toolbar and then on the object with the attributes you want to use.
3. Now insert drawing objects that are to adopt these attributes.

**To transfer drawing attributes from one drawing object to another:**

1. Select all of the objects with the drawing attributes that you want to change.
2. Click on **Adopt Attributes** on the **Drawing Tools** toolbar and then on the object with the attributes you want to use.

**To transfer drawing attributes from one drawing object or some other element to another:**

1. Select the element with the drawing attributes that you want to transfer to a different element.
2. Select **Transfer Format** from the **Standard** toolbar.
3. Now click on the element that you want to receive the drawing attributes. To edit several elements within a rectangular range, drag your mouse to form a rectangle around them.
4. Deselect **Transfer Format** in the **Standard** toolbar.

### Changing Line Style and Line Width

1. Select the objects with the lines or border you would like to change.
2. On the **Drawing Tools** toolbar, click on the **Line Width** or **Line Style** icon and select the desired width or style from the menu.

### Changing the Color of a Line, Arc, Label or Border

1. Select the objects with the line color you would like to change.
2. Click on **Line Color** on the **Drawing Tools** toolbar and select a color. You can change the text color for labels in the **Text Color** menu on the **Format** toolbar.

### Inserting Symbols and Formatting Labels

In all text that you can place in tables and diagrams, as well as in labels, which you insert into documents, you can use a series of control sequences to insert Greek symbols into the text or to format individual text passages. The control sequences always consist of a \ character, followed by a letter. The upper case letter activates the corresponding attribute, and the same letter in lower case deactivates it. You can combine the attributes any way you like.

Control Sequences	Meaning
\S ... \s	Selects the symbol character set for displaying the included text.
\U ... \u	Displays the included text as superscript.
\L ... \l	Displays the included text as subscript.
\I ... \i	Displays the included text as italic.
\B ... \b	Displays the included text as bold.
\CRRGGBB ... \c	Displays the included text in the specified color. Enter the color as a hexadecimal RGB value. For example, \CFF0000Rot\c outputs "Red" as 100% red.
\FRRGGBB ... \F	Displays the included text with the specified background color.
\N	Cancels all of the above conditions and restores the normal appearance of the text.
\\	Displays a \ character. The character has to be entered twice so that it is not interpreted as a control sequence.

**Tip:** You can use the Windows **Character Map** program in the **System Tools** subfolder of the **Accessories** folder to select the Greek symbols.

---

### Changing Line and Fill Transparency

1. Select all of the objects with the line and fill transparency you would like to change.
2. From the **Selection** menu, point to **Line Transparency** or **Fill Transparency** and then select the desired degree of transparency from the submenu.

---

**Note:** The higher the transparency, the more transparent hidden elements will become.

---

## 6.7 Worksheets

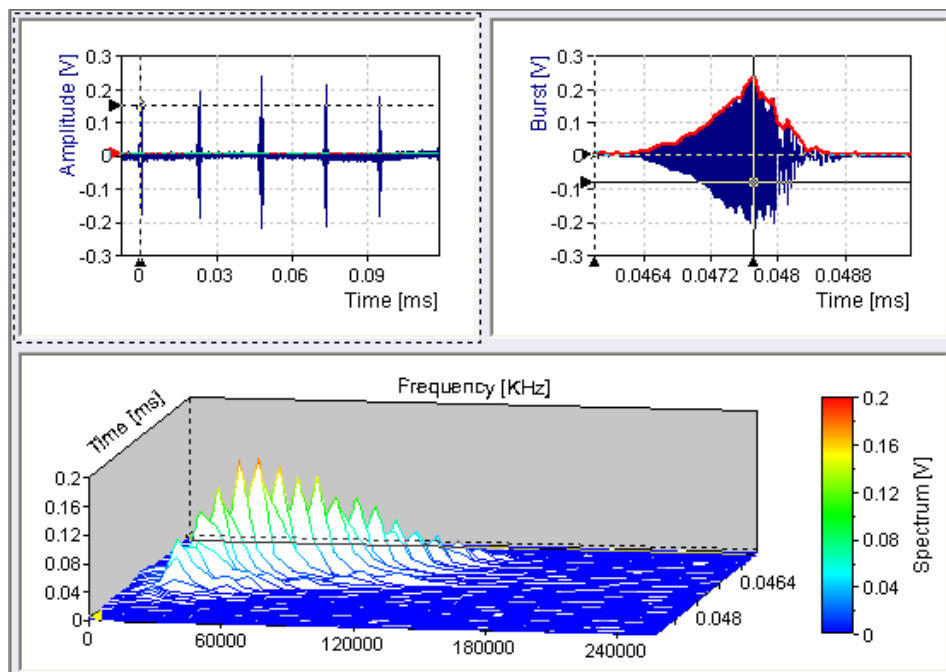
In the worksheet you can arrange presentation objects, i.e. diagrams, text and tables, in panes, and you can set the number of panes. The curves in the diagrams are measured using cursors. As in a document, you can also use linked and embedded objects in a worksheet.

The cursor operating options are completely identical in a worksheet and in a document. Unlike a document, which is oriented to paper output, a worksheet is the ideal environment for working on screen. The panes in the worksheet are connected seamlessly to each other and make efficient use of the screen. The size of the worksheet is not oriented to a page layout and can therefore take up the whole screen. If you display several panes in a worksheet, you can temporarily maximize any one of them. This pane then has the complete window at its disposal so that you can see the contents better.

If you work with object linking, you can use documents and worksheets together. Set up the diagrams to be edited in the project database and insert links to them in a document and a worksheet. Now use the worksheet to move the cursors and the document to print the diagrams.

If you add multi-page diagrams to the worksheet, then they are displayed completely on one page. For multi-page column tables, only the first page is displayed.

The following illustration shows a typical worksheet:



## Working with Worksheets

### Creating a Worksheet

1. In the Folders window, select the folder where you want the new worksheet to be placed.
2. Select the data sets and/or diagrams that are to be displayed in the worksheet in the **Object List**. If you do not select anything, an empty worksheet will be created into which you can insert diagrams later.
3. Click on **Insert Worksheet** in the **FlexPro Objects** toolbar.
4. The Worksheet Wizard appears to guide you through the remaining steps.



### **Creating a Worksheet for a Data Set with a Broad Overview**

Do the following to create a worksheet with two panes, where the upper pane displays an entire data set and the lower pane always displays the enlarged section of the data set located between the two cursors in the upper pane:

1. In the Folders window, select the folder where you want the new worksheet to be placed.
2. Select a data set from the **Object List** that is to be displayed in the worksheet.
3. Click on **Insert Worksheet** in the **FlexPro Objects** toolbar to activate the Worksheet Wizard.
4. Select a layout with two panes.
5. In the new worksheet, move the cursors in the upper pane to display a section enlarged in the lower pane.

### **Linking and Embedding Presentation Objects**

#### **Inserting a Link to a Presentation Object Set Up in the Project Database**

Drag the object with the mouse from the Object List directly to the pane where it should appear.

#### **Embedding a New Presentation Object into the Worksheet**

1. In the Object List, select the data sets that you want displayed in the object.
2. Right-click with your mouse on the pane in the worksheet where the new object is to be added.
3. In the context menu, point to **New**.
4. Select the object type, such as diagram. The Wizard appears to guide you through the rest of the creation process.

#### **Displaying a Data Set or Multiple Data Sets in a New Embedded Diagram**

Simply drag the data sets from the Object List into an empty worksheet pane. Depending on the data structure of the data sets, there are various display options from which you can choose. A new embedded diagram is created in which the data sets are displayed as curves.

### **Embedding a Copy of an Existing Presentation Object into the Worksheet**

While holding down the CTRL key, use your mouse to drag the object from the Object List or a document directly onto the pane where it is to appear.

### **Moving an Existing Presentation Object into the Worksheet**

While holding down the ALT key, use your mouse to drag the object from the Object List or a document directly onto the pane where it is to appear.

---

**Note:** If you want to copy or move an object from a different worksheet, you have to use the clipboard.

---

### **Adding or Removing Panes**

#### **To add a pane via the toolbar:**

- Click on **Add Pane** in the **Worksheet** toolbar.

#### **To remove a pane via the toolbar:**

- Select the pane and then click on **Remove Pane** in the **Worksheet** toolbar.

#### **To split a pane horizontally using the mouse:**

1. Move the mouse to the upper right corner of the pane until the mouse arrow becomes a horizontal split arrow.
2. Now drag the split bar downward to the desired position.

#### **To split a pane vertically using the mouse:**

1. Move the mouse to the lower left corner of the pane until the mouse arrow becomes a vertical split arrow.
2. Now drag the split bar to the right to the desired position.

#### **To remove a pane using the mouse:**

- Move a horizontal or vertical split bar by dragging the mouse so that the pane that is to be removed disappears.

## Changing the Window Layout

To change the number and arrangement of panes in a worksheet, use the **Window Layout** command in the **Worksheet** menu. The **Worksheet Wizard** appears to help you select the layout.

---

**Note:** You can also use your mouse to change the window layout.

---

## Maximizing a Pane

If the worksheet contains several panes, you can display any pane temporarily as maximized by clicking on it and then selecting the **Maximize Pane** option in the **Worksheet** toolbar.

## 6.8 Exporting Presentation Objects, Documents and Worksheets

### Exporting Presentation Objects, Documents and Worksheets as a File

With FlexPro you can export your diagrams, texts, tables, documents and worksheets in a variety of different file formats. The following table provides a summary of options:

Format	Description of Format	Supported Objects
Rich Text (.rtf)*	Text with formatting	Text
Windows Metafile (.wmf)	Vector graphics	2D diagram, 3D diagram, text, column table, cell table, document, worksheet
Windows Enhanced Metafile (.emf)	Enhanced vector graphics	2D diagram, 3D diagram, text, column table, cell table, document, worksheet
Bitmap Graphic (.bmp)	Bit-mapped image	2D diagram, 3D diagram, text, column table, cell table, document, worksheet
JPEG Graphic (.jpg)	Compressed image	2D diagram, 3D diagram, text, column table, cell table, document, worksheet
PNG Graphic (*.png)	Compressed image	2D diagram, 3D diagram, text, column table, cell table, document, worksheet
HTML Document (.htm)	File format of the World Wide Web. Can be published on the Internet/intranet and is displayed by web browsers.	All objects, including complete project databases.

Format	Description of Format	Supported Objects
Text (.txt)*	Text in ANSI code	Text, column table, cell table

The formats identified by \* allow you to store several objects in one single file.

1. Select all of the objects in the Object List that you want to export to the file.  
Alternatively, you can also select an object for export that is in a worksheet or document.
2. Select **Export** from FlexPro's **File** menu.
3. Select the target folder and specify the file name for the file.
4. Under **File Type**, select the format for the file.
5. Select any export options for the file.

## Exporting Presentation Objects, Documents and Worksheets with OLE

You can export presentation objects, documents, worksheets and text as OLE objects and thereby embed copies or links to these objects into OLE client applications. FlexPro offers the objects as **FlexPro Project Database Objects**, **Graphics (Windows Metafile)**, **Images (Enhanced Windows Metafile)** and **Device Independent Bitmaps**. Embedding or linking depends on the third-party application into which the file is to be inserted. Here we describe the process using the popular word processor Microsoft Word 2003.

### To create a dynamic link:

1. Select a presentation object, document or worksheet from the **Object List**.
2. Use the **Copy** command in the **Edit** menu to copy the object to the clipboard.
3. Place the cursor in the location in the Word document where the object is to be inserted.
4. Use the command **Paste Special** in Word's **Edit** menu.
5. In the **Paste Special** dialog box, select the **FlexPro Project Database Object** format and select the option **Paste link**.

---

**Note:** You can also copy an open presentation object to the clipboard by using the command **Copy** in the window where the object is displayed. However, elements in the object must not be selected.

---

**To insert a static copy:**

1. Select a presentation object, document or worksheet from the **Object List**.
2. Use the command **Copy** in the **Edit** menu to copy the object to the clipboard.
3. Place the cursor in the location in the Word document where the object is to be inserted.
4. Use the command **Paste Special** in Word's **Edit** menu.
5. In the **Paste Special** dialog box, select the **Device Independent Bitmap** format.

---

**Note:** You can only use the **Project Database Object** format for links and **not for inserting**.

---

## 6.9 Working with Presentation Templates

### Creating a Presentation or Document Template

1. In the Object List select the diagram, table, text or document that you want to save as a template.
2. Now right-click with your mouse on the selected object and choose **Save as Template**.
3. Follow the wizard's next steps and use Help in the wizard if you require additional assistance.

**See Also**

Presentation and Document Templates

Organizing Presentation and Document Templates

Template Databases

### Using a Presentation or Document Template

1. Select the object from the Object List that you would like to present.
2. Select **Diagram**, **Table**, **Text** or **Document** from the FlexPro **New** menu.
3. Follow the steps of the respective **Wizard** and use the wizard's Help if you require additional assistance.

## See Also

Presentation and Document Templates

Creating a Presentation or Document Template

Template Databases

## Organizing Presentation and Document Templates

Use the **Organizer** dialog box to delete presentation or document templates or to copy templates from one data base to another.

### Opening the Organizer dialog box and selecting the the template type

1. In the FlexPro **Tools** menu, select **Organizer**.
2. In the **Organizer** dialog box, click on the **Templates** tab.
3. In the **Template type** list box select one of the following template types: **Document**, **Diagram**, **Cell Table**, **Column Table** or **Text**.

### Opening the current project database or a template database for organization

- Choose a template database in one of the **Items available in** list boxes.

### Opening a shared template database or any other database for organization

1. Click on one of the **Close Project Database** buttons to close the database currently displayed in the list.
2. Click on **Open Project Database** and select any project database on your hard disk or on the network.

### Copying templates from one project database to another

1. Select the templates you would like to copy.
2. Click **Copy**.

### Deleting templates from a project database

1. Select the template that you would like to delete.
2. Click **Delete**.

## 6.10 Hyperlinks

FlexPro allows you to link objects using hyperlinks. A hyperlink is displayed as a highlighted section of text. If you click on the hyperlink, the event occurs that is associated with that hyperlink:

Type of Hyperlink	Action
Hyperlink to a FlexPro object in the project database	The object is opened.
Hyperlink to a file on the disk or on the network	The file is opened by the program associated with that file type.
Hyperlink to a Website	The page is displayed in the web browser.
A hyperlink that sends an e-mail	A blank e-mail to the specified address is provided.

Hyperlinks can be inserted anywhere that you can have labels, such as the axis label for a diagram, the title of a table or in a text field in a document. Currently, hyperlinks cannot be used in text objects. When exporting to HTML, these hyperlinks are converted into HTML and remain active.

In addition to the hyperlinks in labels, you can specify a hyperlink in the **General** tab of the **Properties** dialog box of any FlexPro object. This hyperlink is generally used as a reference to documentation of the object. When exporting to HTML format, these hyperlinks are incorporated into the heading of the HTML page.

### Working with Hyperlinks

#### Inserting a Hyperlink into a Document

1. Click on the **Label** button on the **Drawing Tools** toolbar.
2. Click on the location where text with a hyperlink is to be inserted.
3. In the dialog box that appears, place the cursor in the edit box.
4. You can now enter a text and select the section to be highlighted as a hyperlink. Alternatively, you can leave the edit box blank. The selected text should not contain any fields.
5. Click on **Insert or Edit a Hyperlink**. A dialog box appears where you can enter the hyperlink.
6. Select one of the four hyperlink buttons, depending on the type of hyperlink you want, and enter the address.
7. Close the **Edit Hyperlink** dialog box by pressing ENTER. The hyperlink has now been entered into the **Edit Text** dialog box.

8. The hyperlink consists of two tags: `<A href="Address">` and `</A>`. The text to be highlighted in the document is located between these two tags. You are free to change this text.

### Adding a Hyperlink to a Diagram or Table

1. Double-click on the text element to which you want to add a hyperlink, such as to an axis label of a diagram or the cell of a cell table.
2. In the **Properties** dialog box that appears, select the passage in the **Text** field that you want to highlight as a hyperlink. Alternatively, you can place the cursor at the point where you want to insert the hyperlink. The selected text should not contain any fields.
3. Click on **Insert or Edit a Hyperlink**. A dialog box appears where you can enter the hyperlink.
4. Select one of the four hyperlink buttons, depending on the type of hyperlink you want, and enter the address.
5. Close the **Edit Hyperlink** dialog box by pressing ENTER. The hyperlink has now been entered into the Properties dialog box.
6. The hyperlink consists of two tags: `<A href="Address">` and `</A>`. The text to be highlighted in the diagram is located between these two tags. You are free to change this text.

### Assigning a Hyperlink to a FlexPro Object

1. Right-click with your mouse on the object in the **Object List** and select **Properties**.
2. Open the **General** tab of the Properties dialog box.
3. Depending on the hyperlink you want, click on one of the four buttons next to the **Hyperlink** input field and enter the address.
4. Close the **Edit Hyperlink** dialog box by pressing ENTER. The hyperlink has now been entered into the **Hyperlink** input field.

### Following a Hyperlink

- To follow a hyperlink in a label such as the axis label of a diagram, simply click on the highlighted text.
- To follow a hyperlink that you created in the **General** tab of a FlexPro object Properties dialog box, use your right mouse button to click on the object in the Object List and select the **Follow Hyperlink** command.



## 6.11 Formatter

Formatters control the output of numbers, angles, strings, calendar time values and time spans. They always start with a '%' character, which is followed by a code, such as %5.3g.

### Standard Formatter

The standard formatters %? and %&? output any data types in a standard format. Floating point values are output using a number of valid places that can be set on the **General** tab in the Properties dialog box of the FlexPro project database. If you use a variant with the & character, data with a unit is output with the unit appended.

### Numeric Value

This formatter controls how numeric data is displayed. You can choose from a variety of formatting options. You can align the digits in a field, specify whether an exponent should be displayed and if so, in which format and basis, specify the number of places after the decimal point, and much more.

#### Syntax

**%**[Attribute][Field size][.Places]Formatter

The syntax of the numeric value formatter consists of the following elements:

Element	Description
Attribute	One or more of the following attribute characters:
	0            Fills up the field size with leading zeros.
	&           Outputs quantities with an appended unit.
	#           Outputs the decimal separator even if no fractional digits are output.
	+ or space   + also outputs the positive polarity sign. A space outputs a space instead of +.
	-           Places the formatted number aligned to the left and fills the remaining field width with spaces.
	'           Outputs and groups the integral part of the number using the 1000th separator for easier legibility.
Field size	Specifies with size (width) of the field into which the formatted number is to be placed as either left or right aligned. If necessary, it is filled with spaces. The entire number is always output, even if the Field size was not specified or a smaller size was specified. If applicable, the appended unit is ignored for the alignment and is

Element	Description														
Places	instead placed after the aligned numeric value.  Depending on the formatter chosen, this selects the number of valid places or fractional digits. If you omit the element, the value preset on the <b>General</b> tab of the project database <b>Properties</b> dialog box is used.														
Formatter	<table><tr><td>f</td><td>Never includes the exponent.</td></tr><tr><td>E, e *</td><td>Always includes the mantissa and exponent.</td></tr><tr><td>G, g *</td><td>An exponent is only used if it leads to a more compact presentation.</td></tr><tr><td>N, n *</td><td>An exponent to the base e is always output.</td></tr><tr><td>B, b *</td><td>An exponent to base 2 is always output.</td></tr><tr><td>M, m *</td><td>If the mantissa equals 1, then only the exponent is output without the mantissa. Otherwise, only the mantissa is output without the exponent. This display is suited for scaling logarithmic axes, for instance.</td></tr><tr><td>F</td><td>Instead of an exponent, an SI prefix is appended, e. g. 1.234 k instead of 1.234 E+003.</td></tr></table> * Use lower case letters to output the exponent in superscript. Example: 10 <sup>3</sup> or 2 <sup>3</sup> . Otherwise, E+003 or B+003 will be output, for instance.	f	Never includes the exponent.	E, e *	Always includes the mantissa and exponent.	G, g *	An exponent is only used if it leads to a more compact presentation.	N, n *	An exponent to the base e is always output.	B, b *	An exponent to base 2 is always output.	M, m *	If the mantissa equals 1, then only the exponent is output without the mantissa. Otherwise, only the mantissa is output without the exponent. This display is suited for scaling logarithmic axes, for instance.	F	Instead of an exponent, an SI prefix is appended, e. g. 1.234 k instead of 1.234 E+003.
f	Never includes the exponent.														
E, e *	Always includes the mantissa and exponent.														
G, g *	An exponent is only used if it leads to a more compact presentation.														
N, n *	An exponent to the base e is always output.														
B, b *	An exponent to base 2 is always output.														
M, m *	If the mantissa equals 1, then only the exponent is output without the mantissa. Otherwise, only the mantissa is output without the exponent. This display is suited for scaling logarithmic axes, for instance.														
F	Instead of an exponent, an SI prefix is appended, e. g. 1.234 k instead of 1.234 E+003.														

## Indexed Text

This formatter offers you the option of assigning texts to numeric values starting at 0, which are then displayed instead of the numbers. You can thus label the columns of a column chart, for instance, with individual text. Do not confuse this formatter with the string formatter, which cannot display any numeric data.

### Syntax

`%("Text0"[, "Text1"[, ..., "Textn"]])`

The syntax of the formatter for indexed text consists of the following elements:

Element	Description
Text <sub>0</sub> ... Text <sub>n</sub>	The text to be output instead of the values 0 through n.

## Calendar Time

FlexPro saves calendar times as numeric values that contain the number of seconds

since January 1, 1970 in "Universal Time Coordinated" (UTC). Using a calendar time formatter, you can display these numeric values as a calendar time. FlexPro converts the UTC values automatically to the time zone set in Windows Regional and Language Options. You can, however, specify an option that is to result in output in universal time coordinated. The formatter specifies which time value elements should be displayed. There is a corresponding formatting element for each of the elements listed below. This formatting element starts with a % character. You can write any text between the individual formatting elements, where a percent sign has to be entered as %%.

### Syntax

**%[.Fractional digits]([UTC] Date)**

The syntax of the calendar timeformatter consists of the following elements:

Element	Description																														
Fractional digits	Selects the maximum number of fractional digits for seconds. If you omit the element, then the default value used is 6, which is equivalent of a 1 $\mu$ s resolution.																														
UTC	Outputs the date as Universal Time Coordinated (UTC).																														
Date	Any text in which the following element formatters can be embedded: <table> <tr> <td>Name of the day of the week, abbreviated</td><td>%a</td></tr> <tr> <td>Name of the day of the week</td><td>%A</td></tr> <tr> <td>Name of the month, abbreviated</td><td>%b</td></tr> <tr> <td>Name of the month</td><td>%B</td></tr> <tr> <td>Date and time, based on Windows Region and Language Options settings</td><td>%c</td></tr> <tr> <td>Day of the month in decimal format (00 - 31)</td><td>%d</td></tr> <tr> <td>Hour in 24-hour format (00 - 23)</td><td>%H</td></tr> <tr> <td>Hour in 12-hour format (01 - 12)</td><td>%I</td></tr> <tr> <td>Day of the year as a decimal number (001 - 366)</td><td>%j</td></tr> <tr> <td>Month as a decimal number (01 - 12)</td><td>%m</td></tr> <tr> <td>Minute as a decimal number (00 - 59)</td><td>%M</td></tr> <tr> <td>A.M./P.M. indicator, based on Windows Region and Language Options settings</td><td>%p</td></tr> <tr> <td>A.M./P.M. indicator</td><td>%P</td></tr> <tr> <td>Second as a decimal number (00 - 59)</td><td>%S</td></tr> <tr> <td>Calendar week, starting with Sunday, as a decimal number</td><td>%U</td></tr> </table>	Name of the day of the week, abbreviated	%a	Name of the day of the week	%A	Name of the month, abbreviated	%b	Name of the month	%B	Date and time, based on Windows Region and Language Options settings	%c	Day of the month in decimal format (00 - 31)	%d	Hour in 24-hour format (00 - 23)	%H	Hour in 12-hour format (01 - 12)	%I	Day of the year as a decimal number (001 - 366)	%j	Month as a decimal number (01 - 12)	%m	Minute as a decimal number (00 - 59)	%M	A.M./P.M. indicator, based on Windows Region and Language Options settings	%p	A.M./P.M. indicator	%P	Second as a decimal number (00 - 59)	%S	Calendar week, starting with Sunday, as a decimal number	%U
Name of the day of the week, abbreviated	%a																														
Name of the day of the week	%A																														
Name of the month, abbreviated	%b																														
Name of the month	%B																														
Date and time, based on Windows Region and Language Options settings	%c																														
Day of the month in decimal format (00 - 31)	%d																														
Hour in 24-hour format (00 - 23)	%H																														
Hour in 12-hour format (01 - 12)	%I																														
Day of the year as a decimal number (001 - 366)	%j																														
Month as a decimal number (01 - 12)	%m																														
Minute as a decimal number (00 - 59)	%M																														
A.M./P.M. indicator, based on Windows Region and Language Options settings	%p																														
A.M./P.M. indicator	%P																														
Second as a decimal number (00 - 59)	%S																														
Calendar week, starting with Sunday, as a decimal number	%U																														

Element	Description
	(00 - 53)
	Calendar week, starting with Monday, as a decimal number (00 - 53) %W
	Day of the week as a decimal number (0 - 6; Sunday is 0) %w
	Date, based on Windows Region and Language Options settings %x
	Time, based on Windows Region and Language Options settings %X
	Year, two-digit, as a decimal number (00 - 99) %y
	Year, four-digit, as a decimal number (00 - 99) %Y
	Name of the time zone, abbreviated %z
	Name of the time zone %Z
	Percent sign %%

### Examples

%(%A, the %th of %B %Y %H:%M)	Outputs, for instance "Tuesday, the 16th of February 2010 17:15"
%(UTC%A, the %th of %B %Y %H:%M UTC)	Outputs, for instance "Tuesday, the 16th of February 2010 04:15:00 PM"
%.0(%H:%M:%S) or %.0(%X)	Outputs only the time, so the seconds are output without fractional digits.

### Timespan

Timespans can be formatted as H:M:S or D:H:M:S. The number of fractional digits as well as the number of places for the hour or day can be adjusted.

### Syntax

%[Places][.Fractional digits]Formatter

The syntax of the timespanformatter consists of the following elements:

Element	Description
Places	Specifies the number of digits for the day or hours. If necessary, it is filled with leading zeros.

Element	Description
Fractional digits	Selects the maximum number of fractional digits for seconds. If you omit the element, then the default value used is 6, which is equivalent of a 1 $\mu$ s resolution.
Formatter	t            Output as H:M:S.
	D            Output as D:H:M:S.

### Examples

%3.2t	Formats the timespan as HHH:MM:SS.ss.
%.0T	Formats the timespan as D:HH:MM:SS. The seconds are output without fractional digits.

### Angle

Angles can be displayed in degrees, minutes, seconds ( $^{\circ}$  ' ') or as a fraction of  $\pi$ . The existing angle data can be formatted in degrees or radians.

### Syntax

%[.Places]Formatter

or

%[.Denominator]Formatter

The syntax of the angleformatter consists of the following elements:

Element	Description
Places	When outputting in degrees, minutes, seconds, the maximum number of digits is specified for the seconds. If you omit the element, the default value 6 is used.
Denominator	When outputting as a fraction of $\pi$ , this specifies the denominator of the fraction. If you omit the element, then instead of outputting a fraction, a decimal number with the maximum number of fractional digits predefined on the <b>General</b> tab of the project database <b>Properties</b> is output.
Formatter	r, R            Outputs degrees, minutes, seconds.
	p, P            Outputs a fraction of $\pi$ . Use upper case letters if the data to be formatted is in radians (0 to $2\pi$ ) and lower case letters if the data is in degrees (0 to 360).

**Examples**

- `%2r`                      Outputs, for instance, "15° 12´ 24,34´´".
- `%4P`                      Outputs, for instance, "3/4  $\pi$ ".
- `%P`                        Outputs, for instance, "0.75  $\pi$ ".

**String**

You can limit strings to a maximum length or fill them up to a minimum length. The text can be displayed as left or right justified.

**Syntax**

`%[-][MinFieldSize][.MaxFieldSize]s`

or

`%[-][MinFieldSize][.MaxFieldSize]S`

The syntax of the string formatter consists of the following elements:

Element	Description	
-	If specified, the field alignment is left-justified; otherwise, it is right-justified.	
MinFieldSize	Specifies the minimum field size in characters. The value is only relevant if the string is shorter than the specified value. It is then filled with spaces.	
MaxFieldSize	Specifies the minimum field size in characters. The value is only relevant if the string is longer than the specified value. Excess characters are then cut off.	
Formatter	S	Output with double <code>\</code> character. This prevents the <code>\</code> character in the text from being interpreted as the start symbol in control sequences, such as <code>\B</code> for bold.
	s	Output unchanged. Control sequences are interpreted and taken into account in the text formatting.

**Examples**

- `%s`                        Outputs, for instance, "Test".
- `%3s`                      Outputs, for instance, "Tes".

%6s                      Outputs, for instance, " Test".

## 6.12 Field and Placeholder

A field consists of an FPScript expression that supplies a value or string to be displayed and a formatter that determines how the data is to be formatted for output.

You can insert fields in any text, for instance, in axis labels, text objects, document labels, etc. Values to be displayed can be calculation results, header information such as a data set name and physical unit, or the current date and time.

The following field shows, for instance, the first value in the X component of a signal with three fractional digits:

```
%.{3f}{Signal.x[0]}
```

In this example, `Signal.x[0]` is the FPScript expression that supplies the value, and `%.{3f}{...}` is the formatter that determines that the number is to be formatted with three fractional digits instead of an exponent. Formatting information is optional here; the simple formatter `%{...}` outputs any data types in a standard format. Floating point values are output using a number of valid places that can be set on the **General** tab in the Properties dialog box of the FlexPro project database.

Fields always start with a % sign. Therefore, if you want to insert a single percent sign into a text, you need to type in the percent sign twice to prevent it from being interpreted by the formatter. For instance:

```
The probability of error is %{Probability}%%.
```

### Accessing Object Attributes in Fields

Since FPScript has full access to FlexPro's Automation object model, you can display any object attributes, also known as properties. The pre-defined fields for axis labeling in diagrams is used frequently here, for instance, to determine the names, comments and units of the data sets to be plotted over an axis. The FPScript fields used for this can be somewhat long, which is why FlexPro uses **placeholders**. Placeholders also have a formatter, which is followed by a name instead of FPScript code. The standard text for labeling the Y axis of a 3D curve, for instance, looks like this:

```
%<NameOrComments> %<[Unit]>
```

FlexPro replaces both placeholders when text is created using the following FPScript fields:

```
%{.DataSets(%<DataSet>).YValueObject.NameOrComments(.DataSets(%<DataSet>).YComponent)}
```

FPScript defines a standard object, which can be activated using the optional keyword

`ThisObject`. The expression `.DataSets` is thus equivalent to the expression `ThisObject.DataSets`. When evaluating the axis label of a curve, `ThisObject` is a reference to the curve. With `ThisFPObject`, you have access to the FlexPro object, such as the diagram or table containing the field.

The fields are evaluated for every series of the 3D curve. `%<DataSet>` is a placeholder that is replaced before the evaluation by the index of the corresponding series.

The FPScript expression accesses the `DataSets` collection of the curve in order to access the `NameOrComments` or `UnitInBrackets` property of the data set displayed as a series of the curve.

The following table specifies which objects `ThisObject` and `ThisFPObject` reference:

Field contained in	ThisObject points to	ThisFPObject points to
Axis labeling of an axis	Axis	Diagram
Axis labeling, curve labeling or legend entry of a curve	Curve	Diagram
Legend title	Legend	Diagram
Color legend label	Color Legend	3D diagram
Column table title	Column table	Column table
Column title in a column table	Table column	Column table
Cell table title	Cell table	Cell table
Cell in a cell table	Cell	Cell table
Text object	Text object	Text object
Label in a document/diagram	Label	Document/diagram

Special Placeholders

FlexPro also uses some placeholders that are not followed by FPScript code:

Placeholder	Used in	Meaning
<code>%&lt;DataSet&gt;</code>	Axis label and legend entry for 3D curve	Represents the number of the respective data series in the curve for which an axis label or legend entry is created and is used as an index for the <code>Curve3DDataSets</code> collection of the <code>Curve3D</code> object.
<code>%&lt;Index&gt;</code>	"	Represents the number of respective data series in a 3D data set for which an axis label or legend entry is created. Used for displaying the data series index.
<code>%&lt;[Index]&gt;</code>	"	Like <code>%&lt;Index&gt;</code> , but the output result is in



Placeholder	Used in	Meaning
%<YValue> %<XValue> %<ZValue>	Data label and marker in a 2D or 3D curve	brackets. The values of the individual components of the data point to be labeled. %<ZValue> can only be used in 3D curves.
%<YUnit> %<XUnit> %<ZUnit>	"	The physical values of the individual components of the data point to be labeled. %<ZUnit> can only be used in 3D curves.
%<DataSetValue>	Data labeling in a 2D or 3D curve	The value of the additional data set for data labeling. The value is assigned to the data point to be labeled.
%<XIndex>	Data labeling in a 3D Curve	The row index of the data point to be labeled in a 3D data set.
%<ZIndex>	Data labeling in a 3D Curve	The column index of the data point to be labeled in a 3D data set.
%<PageNumber>	Text drawing object in a document	The page number in a document where the text drawing object is located.

### Fields for Outputting Complex Numbers

If the data set with the contents you want to display contains complex numbers, you can use the following functions in the field:

Function	Meaning
<b>RealPart</b> (DataSet)	Displays the real part of the complex number in the data set.
<b>ImaginaryPart</b> (DataSet)	Displays the imaginary part of the complex number in the data set.
<b>Absolute</b> (DataSet)	Displays the absolute value of the complex number in the data set.
<b>Phase</b> (DataSet)	Displays the phase of the complex number in the data set.

### Indexing Scalar Values for Display

If the data set with the value that is to be displayed does not contain a scalar value, but instead contains a data series or data matrix, you can use an Index to display a particular value. If you do not use an index, the first value is taken. If you enter the number -1 as the index, the last value is always taken.

If the data set contains a signal, signal series or space curve, you can select the desired component and append .X, .Y or .Z to the name. If you do not specify a component, the Y component is used.

## Examples

<code>g{Signal.Y[10]}</code>	Displays the 11th value of the Y component of the signal.
<code>%g{Signal.X[-1]}</code>	Displays the last value of the X component of the signal.
<code>%g{DataMatrix[0][10]}</code>	Displays the 11th value of the 1st column in the data matrix.

## Fields with a User Prompt

You can use special fields to embed user prompts into the text. Use this option to prompt input from the user, for instance, when documents are updated for subsequent display in the document.

There are two options for embedding a user prompt into a text. You can either build a user prompt into the FPScript expression of a field, such as `%{TextInput("Please enter your name")}`, or you can use a special formatter that requests input, such as `%"Recorder:\DM2000\MX2000"`. The second option has various advantages. First, you can specify default values for input, such as "DM2000" or "MX2000". These are then displayed in a drop list box. Second, FlexPro combines several user prompts embedded in the same text, and third, FlexPro remembers past entries so that you only have to enter changes when making revisions.

## 7 Customizing FlexPro

### 7.1 Customizing the User Interface

The FlexPro user interface consists of a menu bar and several toolbars, where, depending on the working environment, i.e., depending on which window is currently active in the workspace, a particular menu bar and a specific selection of toolbars are displayed.

#### Menus and Toolbars

A menu shows a list of commands. In addition to some commands, there are symbols to help you quickly associate a command with the symbol. Most menus are located on the menu bar. This is the toolbar at the very top of the screen. Toolbars can contain buttons, menus or a combination of both.

You can customize the menus and toolbars to your needs. You can add or delete buttons and menus on toolbars, create your own custom toolbars, or show, hide or move toolbars. The menu bar can also be customized just like any toolbar. For instance, you can quickly add buttons and menus to the menu bar or remove them, but you cannot hide the menu bar.

#### Shortcuts

You can perform frequent tasks faster when you use shortcuts, which means that you need to press one or more keys to perform a task. For instance, using a shortcut you can save the active project database by pressing the keys CTRL+S, by clicking on **Save Project Database** on the **Standard** toolbar or in the **File** menu.

You can customize shortcuts in FlexPro. You can assign commands to shortcut key combinations that have not yet been assigned as a shortcut, and delete unnecessary shortcuts. You can undo settings that you have made at any time by resetting the shortcuts to the default settings.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

## Customizing Menus and Toolbars

### Adding a Button, Menu or Command

Carry out one or more of the following actions:

#### Adding a Button, Menu or Command to a Toolbar

1. In the **Tools** menu, click on **Customize**.
2. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box, click on the **Toolbars** tab.
  2. Check the box for the toolbar you want to display.
3. Carry out one of the following actions:

##### Adding a Button or an Built-In Menu to a Toolbar

1. Click on the **Commands** tab.
2. Click on the category in the **Categories** field for the command that is to be executed or for the menu that is to be opened using the button.
3. Drag the desired command, menu or macro from **Commands** to the toolbar displayed.

##### Adding a Customized Menu to a Toolbar

1. Click on the **Commands** tab.
  2. Under **Categories** click on **New Menu**.
  3. Drag **New Menu** from **Commands** to the toolbar displayed.
  4. Use your right mouse button to click on the new menu on the toolbar and select **Button Appearance** from the context menu.
  5. In the **Button Appearance** dialog box, enter the name of the menu in the **Button text** field.
4. In the **Customize** dialog box, click on **Close**.

#### Adding a Command to a Menu

1. In the **Tools** menu, click on **Customize**.
2. If the menu to be changed is located on a toolbar, you have to display the toolbar:
  1. In the **Customize** dialog box, click on the **Toolbars** tab.

2. Check the box for the toolbar that you want to display.
3. In the **Customize** dialog box, click on the **Commands** tab.
4. Under **Categories** click on the category for the command.
5. Drag the command you want from **Commands** to the menu you want to change without releasing the mouse button. When a list of commands appears in the menu, point to the location where the command is to be inserted and then release the mouse button.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

## Deleting a Button or Menu Command

### Deleting a Button

1. In the **Tools** menu, click on **Customize**.
2. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box, click on the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. Drag the button you want to delete off of the toolbar.
4. In the **Customize** dialog box, click on **Close**.

### Deleting a Command from a Menu

1. In the **Tools** menu, click on **Customize**.
2. Click the menu that contains the command you want to delete.
3. Drag the command you want to delete off of the menu.
4. In the **Customize** dialog box, click on **Close**.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

## Moving or Copying Buttons, Menus or Menu Commands

### Moving or Copying a Menu Command

1. In the **Tools** menu, click on **Customize**.

2. Make sure the toolbars you want to change are visible:
  1. In the **Customize** dialog box, click on the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. Click on the menu with the command that you want to move or copy.
4. Carry out one of the following actions:
  - To move the command, drag it to rest over the menu you want to change. When the menu displays a list of commands, point to where you want the command to appear, and then release the mouse.
  - To copy the command, hold down CTRL and drag the command to rest over the menu you want to change. When the menu displays a list of commands, point to where you want the command to appear, and then release the mouse.
5. In the **Customize** dialog box, click on **Close**.

### **Moving or Copying a Menu on a Toolbar**

1. In the **Tools** menu, click on **Customize**.
2. Make sure the toolbars you want to change are visible:
  1. In the **Customize** dialog box, click on the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. Carry out one of the following actions:
  - To move a menu, click on it and then drag it to a new location either on the same toolbar or on another toolbar.
  - To copy a menu, hold down the CTRL button and drag the menu to the new location.
4. In the **Customize** dialog box, click on **Close**.

### **Moving or Copying a Toolbar Button**

1. In the **Tools** menu, click on **Customize**.
2. Make sure the toolbars you want to change are visible:
  1. In the **Customize** dialog box, click on the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. Carry out one of the following actions:
  - To move a toolbar button, hold down the ALT key and then drag the button to the

new location.

- To copy a toolbar button, hold down CTRL+ALT and then drag the button to the new location.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

### Assigning or Changing a Shortcut Key

You can select a menu command or a button using a shortcut key by pressing CTRL+<a letter>. For instance, to open the **Edit** menu, press ALT+E.

Carry out one of the following actions:

1. In the **Tools** menu click **Customize**.
2. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box, switch to the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. In the **Customize** dialog box, switch to the **Commands** tab.
4. With the **Customize** dialog box open, carry out one of the following actions:

### Assigning or Changing a Shortcut Key for a Toolbar Button

1. Right-click with your mouse on the button on the toolbar and select **Button Appearance** from the context menu.
2. In the **Button Appearance** dialog box select **Text Only** or **Image and Text**, and in the **Button text** field give the toolbar button a name, adding an ampersand (&) before the letter that you want to use as the shortcut key.
3. Press the ENTER key.

### Assigning or Changing a Shortcut Key for a Menu Command

1. Click on the menu name on the menu bar or on the toolbar to open the menu.
2. Right-click with your mouse on a menu command in the menu and select **Button Appearance** from the context menu.
3. In the **Button Appearance** dialog box select **Text Only** or **Image and Text**, and in the **Button text** field give the toolbar button a name, adding an ampersand (&) before the letter that you want to use as the shortcut key.
4. Press the ENTER key.

5. In the **Customize** dialog box, click **Close**.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

### Creating a Custom Toolbar

1. In the **Tools** menu, click on **Customize**.
2. Click on the **Toolbars** tab.
3. Click on **New**.
4. Enter the name you want to use in the **Toolbar name** field.
5. Click on the **Commands** tab.
6. Click on the category you want under **Categories**.
7. Drag the command you want from the **Commands** field to the toolbar displayed.
8. After adding all of the buttons and menus you want, click on **Close**.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

### Grouping Related Buttons and Menus on a Toolbar

You can add a separator bar before the first and after the last item in a group to distinguish the group from other buttons and menus on a toolbar.

1. In the **Tools** menu, click on **Customize**.
2. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box, click on the **Toolbars**.
  2. Check the box for the toolbar that you want to display.
3. To add a separator bar, keep the **Customize** dialog box open, use right-click with your mouse on the toolbar, and then click on **Start Group**. A separator bar is added to the left of the button on a horizontal toolbar, and above a button on a vertical docked toolbar.

---

**Note:** To remove a separator bar between two buttons, drag one button closer to the other.

---



---

**Note:** The user interface of FlexPro View cannot be customized.

---

### **Moving a Submenu to Any Location on the Screen**

1. Make sure the submenu to be moved is visible. For instance, click on the **Drawing Tools** toolbar and then click **Line Color** to open the **Line Color** submenu.
2. Click on the **Move** handle at the top of the submenu to move the submenu to a new location. The submenu becomes a moveable toolbar.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

### **Customizing a Context Menu**

You cannot add, delete or reset context menus in FlexPro. However, you can change a command image, rename a command, display an image or text for a command and move or copy commands between context menus in the same way you would other commands in FlexPro.

1. In the **Tools** menu, click on **Customize**.
2. Carry out one of the following actions:

#### **Deleting a Command from a Context Menu**

1. In the **Customize** dialog box, click on the **Menus** tab.
2. In the **Select context menu** field, select the context menu containing the command.
3. Drag the command out of the menu.

#### **Adding a Command to a Context Menu**

1. In the **Customize** dialog box, click on the **Menus** tab.
2. In the **Context Menu** field, select the context menu that you want to change.
3. In the **Customize** dialog box, click on the **Commands** tab.
4. Under **Categories** click on the category for the command.
5. Drag the command you want from **Commands** to the relevant location in the context menu and then release the mouse button.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

## Renaming a Button or Menu

### Renaming a Button or Menu Command

1. In the **Tools** menu, click on **Customize**.
2. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box, click on the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. Carry out one of the following actions:

#### Renaming a Menu Command

1. Click on the menu that contains the command.
2. Use the right mouse button to click on the command and select **Button Appearance** from the context menu.
3. In the **Button Appearance** dialog box, select **Text Only** or **Image and Text** and then give the command a name in the **Button text** field.
4. Press the ENTER key.

#### Renaming a Toolbar Button

1. Use the right mouse button to click on the button and select **Button Appearance** from the context menu.
  2. In the **Button Appearance** dialog box, select **Text Only** or **Image and Text** and then give the command a name in the **Button text** field.
  3. Press the ENTER key.
4. In the **Customize** dialog box, click on **Close**.

### Renaming a Menu

1. In the **Tools** menu, click on **Customize**.
2. Use your right mouse button to click on the menu that you want to rename and select **Button Appearance** from the context menu.
3. In the **Button Appearance** dialog box, give the menu a name in the **Button text** field.

4. With the **Customize** dialog box open, click on the menu that you want to rename.
5. Press the ENTER key.
6. In the **Customize** dialog box, click on **Close**.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

## Deleting a Toolbar or Menu

### Deleting a Custom Toolbar

1. In the **Tools** menu, click on **Customize** and then click on the **Toolbars** tab.
2. In the **Toolbars** field, select the custom toolbar you want to delete.
3. Click on **Delete**.

---

**Note:** You cannot delete a built-in toolbar. When you select a built-in toolbar, the **Reset** button appears, which restores the toolbar to its default buttons, menus, and submenus.

---

### Deleting a Menu from a Toolbar

1. In the **Tools** menu, click on **Customize**.
2. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box, click on the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. With the **Customize** dialog box open, drag the menu you want to delete off the toolbar.
4. In the **Customize** dialog box, click on **Close**.

---

**Note:** When you delete a built-in menu, the menu is still available in the **Customize** dialog box. However, when you delete a custom menu, it is permanently deleted. To remove a custom menu and save it for later use, create a toolbar for storing unused menus. Move the menu to this storage toolbar, and then hide the storage toolbar.

---

---

**Note:** The user interface of FlexPro View cannot be customized.

---

## Showing or Hiding a Toolbar

### Showing a Toolbar

- Use your right mouse button to click on any toolbar, and then click the toolbar you want to show on the context menu.
- If you do not see the toolbar you are looking for on the shortcut menu, click on **Customize**, click on **Toolbar** tab, and then, in the **Toolbars** list, click the on toolbar you want.

### Hiding a Toolbar

- Use your right mouse button to click on the toolbar, and then clear the check box to the left of the toolbar you want to hide.

---

**Note:** To hide a floating toolbar quickly, click **Close** on the toolbar.

---

### Showing All Buttons on a Toolbar

- Use the move handle on the toolbar to drag the toolbar to its own row. You can also move the toolbar to a location other than the edge of the program window so that the toolbar becomes a floating toolbar.

### Moving a Toolbar

- Drag the move handle at the left edge of a docked toolbar, or the title bar on a floating toolbar, to the new location.

---

**Note:** If you drag the toolbar to the edge of the program window, it becomes a docked toolbar. When you move a docked toolbar, this might affect the location and size of other toolbars on the same row.

---

### Animating Menus

1. In the **Tools** menu, click on **Customize**.
2. Click on the **Menus** tab.
3. In the **Menu animations** list, select the animation you want to use.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

### Resizing a Toolbar

- To resize a floating toolbar, move the pointer over any edge until it changes to a double-headed arrow, and then drag the edge of the toolbar.

---

**Note:** You cannot resize a docked toolbar to a specific size.

---

### Resizing a Toolbar Button or Drop-Down List Box

#### Increasing the Size of Toolbar Buttons

1. In the **Tools** menu, click on **Customize**.
2. Click on the **Options** tab.
3. Check the box **Large icons**.

#### Changing the Width of a Drop-down List Box on a Toolbar

1. In the **Tools** menu, click on **Customize**.
2. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box, click on the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. With the **Customize** dialog box open, click on the list box you want to change, such as the **Font** or **Font Size** field on the **Format** toolbar.
4. Point to the left or right edge of the field. When the pointer changes to a double-headed arrow, drag the edge of the box to change its width.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

## Changing an Image on a Button

### Selecting an Existing Image for a Toolbar Button or Menu Command

1. In the **Tools** menu, click **Customize**.
2. If the image you want to change is on a toolbar, make sure the toolbar is visible:
  1. In the **Customize** dialog box, switch to the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. With the **Customize** dialog box open, carry out one of the following actions:

#### Change a Toolbar Button Image

1. Use the right mouse button to click on the relevant button on the toolbar and select **Button Appearance** from the context menu.
2. In the **Button Appearance** dialog box select **Select user-defined image** and then click on an image.

#### Changing a Menu Command Image

1. Click the menu name on the menu bar or toolbar, or click the context menu that contains the command with the image you want to change.
2. Use the right mouse button to click on the command and select **Button Appearance** from the context menu.
3. In the **Button Appearance** dialog box, select **Select user-defined image** and then click on an image.

### Pasting a Picture Created in a Graphics Program to a Toolbar Button or Menu Command

For a clear image, copy a graphic that is the same size as a built-in button image or a menu command image. The default built-in button image is 16 x 16 pixels. A larger image might be distorted because it will be scaled to fit the size of the button.

1. Copy to the clipboard the image that you want to appear on the toolbar button or menu command. If you can select the format for the copied graphic, select a bitmap or picture format.
2. Switch to FlexPro.
3. In the **Tools** menu click **Customize**.
4. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box, switch to the **Toolbars** tab.

2. Check the box for the toolbar that you want to display.
5. Carry out one of the following actions:

#### **Change a Toolbar Button Image**

Use the right mouse button to click on the relevant button on the toolbar and select **Button Appearance** from the context menu.

#### **Changing a Menu Command Image**

1. Click the menu name on the menu bar or toolbar, or click the context menu that contains the command with the image you want to change.
  2. Use the right mouse button to click on the command and select **Button Appearance** from the context menu.
6. In the **Button Appearance** dialog box, select **Select user-defined image** and then click on **New**.
  7. In the **Edit Button Image** dialog box, click **Paste**.

### **Copying an Image between Toolbar Buttons or Menu Commands**

1. In the **Tools** menu, click **Customize**.
2. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box switch to the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. To copy the image, carry out one of the following actions:

#### **Copying a Button Image**

Right-click with your mouse on the toolbar button with the desired image and select **Copy Button Image** from the context menu.

#### **Copying a Command Image**

Click on the menu that contains the image, right-click with your mouse on the command, and then select **Copy Button Image** from the context menu.

4. To paste an image, carry out one of the following actions:

#### **Pasting a Copied Image on a Toolbar Button**

Right-click with your mouse on the toolbar button with the desired image and select **Paste Button Image** from the context menu.

#### **Pasting a Copied Image in a Menu Command**

Click on the menu with the menu command you want to change, then use your

right mouse button to click on the command and select **Paste Button Image** from the context menu.

### **Editing the Image of a Toolbar Button or Menu Command**

1. In the **Tools** menu click **Customize**.
2. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box switch to the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.
3. To copy the image, carry out one of the following actions:

#### **Copying a Button Image**

Right-click with your mouse on the toolbar button with the desired image and select **Copy Button Image** from the context menu.

#### **Copying a Command Image**

Click on the menu that contains the image, right-click with your mouse on the command, and then select **Copy Button Image** from the context menu.

4. Carry out one of the following actions:

#### **Editing a Toolbar Button Image**

Use the right mouse button to click on the relevant button on the toolbar and select **Button Appearance** from the context menu.

#### **Editing a Menu Command Image**

1. Click the menu name on the menu bar or toolbar, or click the context menu that contains the command with the image you want to change.
  2. Use the right mouse button to click on the command and select **Button Appearance** from the context menu.
5. In the **Button Appearance** dialog box, select **Select user-defined image** and then click on **New**.
  6. In the **Edit Button Image** dialog box, click **Paste** to copy the image currently assigned to the command to the image editor.
  7. Select the desired editing tools to edit the image.

---

**Note:** The user interface of FlexPro View cannot be customized.

---



## **Adding an Image, Text or Both to a Menu Command or Button**

### **Adding an Image with Text or Only Text to a Menu Command**

1. In the **Tools** menu, click on **Customize**.
  2. If the menu command to be changed is located on a toolbar, make sure that the toolbar is showing:
    1. In the **Customize** dialog box, click on the **Toolbars** tab.
    2. Check the box for the toolbar that you want to display.
  3. With the **Customize** dialog box open, click on the menu that contains the command you want to change.
  4. Use your right mouse button to click on the command you want to change and then select the desired option from the context menu.
- 

#### **Note:**

Some commands do not have an assigned image and can only be displayed as text.

A menu command can only be displayed as an image if you copy or move the command to a toolbar.

---

### **Displaying Text, an Image or Both on a Toolbar Button**

1. In the **Tools** menu, click on **Customize**.
  2. Make sure the toolbar you want to change is visible:
    1. In the **Customize** dialog box, click on the **Toolbars**.
    2. Check the box for the toolbar that you want to display.
  3. In the **Customize** dialog box, click on the **Commands** tab.
  4. With the **Customize** dialog box, click on the button you want to change.
  5. Use your right mouse button to click on the toolbar button you want to change and then select the desired option from the context menu.
- 

**Note:** You cannot modify the text or image of a button that opens a list or menu with the click of a mouse.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

### Renaming a Custom Toolbar

You can rename any custom toolbar that you have created. However, you cannot rename a built-in toolbar that comes with FlexPro.

1. In the **View** menu, point to **Toolbars** and then click on **Customize**.
2. Click on the **Toolbars** tab.
3. Under **Toolbars**, click on the custom toolbar that you want to rename.
4. Click on **Rename**.
5. Enter a name for it under **Toolbar name** and then press ENTER.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

### Restoring Original Settings for Buttons, Commands or Toolbars

#### Restoring Original Settings for a Menu

1. In the **Tools** menu, click on **Customize** and then on the **Menus** tab.
2. Click on the name of the menu to be restored under **Show menu for**.
3. Click on **Reset**.

#### Restoring the Default Buttons and Menus on a Built-In Toolbar

1. In the **Tools** menu, click on **Customize** and then click on the **Toolbars** tab.
2. Click on the name of the toolbar to be restored under **Toolbars**.
3. Click on **Reset**.

---

**Note:** You cannot reset a custom toolbar.

---

#### Restoring the Default Settings for a Built-In Toolbar Button or Menu Command

1. In the **Tools** menu, click on **Customize**.
2. Make sure the toolbar you want to change is visible:
  1. In the **Customize** dialog box, click on the **Toolbars** tab.
  2. Check the box for the toolbar that you want to display.

3. With the **Customize** dialog box open, carry out one of the following actions:
  1. To restore a toolbar button, use your right mouse to click on the appropriate button, and then click on **Reset** in the context menu.
  2. To restore a menu command, click the menu that contains the command. Right-click with your mouse on the menu command, and then click **Reset**.

### **Displaying Default Toolbar Buttons and Menu Commands**

1. In the **Tools** menu, click on **Customize** and then click on the **Toolbars** tab.
2. Click on **Reset All**.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

### **Showing or Hiding Shortcut ScreenTips**

1. In the **Tools** menu, click on **Customize**.
2. Click on the **Options** tab.
3. Select or clear the **Show ScreenTips on toolbars** and **Show shortcut keys in ScreenTips** check boxes.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

### **Showing or Hiding Toolbar ScreenTips**

1. In the **Tools** menu, click on **Customize**.
2. Click on the **Options** tab.
3. Select or clear the **Show ScreenTips on toolbars** check box.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

## Customizing Shortcuts

### Assigning or Removing a Shortcut

Shortcuts can be assigned to commands, macros or frequently used symbols.

1. In the **Tools** menu, click on **Customize**.
2. Click on **Keyboard**.
3. Under **Categories** click on the category containing the command or item.
4. Under **Commands** click on the name of the command or item.

Any shortcut keys that are currently assigned appear in the **Current keys** field.

5. Carry out one of the following actions:

#### Assigning a Shortcut Key

1. Begin shortcut key combinations with CTRL, ALT, or a function key.
2. Click on the **Press new shortcut key** field and press the shortcut key combination you want to assign. For example, press ALT plus the desired key.
3. Look at **Currently assigned to** to see if the shortcut key combination is already assigned to a different command or item. If so, select another combination.
4. Click on **Assign**.

#### Deleting a Shortcut Key

1. Under **Current keys** click on the shortcut key combination to be removed.
2. Click on **Delete**.

---

**Note:** If you use a programmable keyboard, the keyboard shortcut CTRL+ALT+F8 may be reserved for initiating keyboard programming.

---

---

**Note:** The user interface of FlexPro View cannot be customized.

---

### Restoring Original Settings for Shortcuts

1. In the **Tools** menu, click on **Customize**.
2. Click on **Keyboard**.
3. Click on **Reset All**.

---

**Note:** The user interface of FlexPro View cannot be customized.

---

## 7.2 User Profile

In your user profile, FlexPro stores several settings that you make while working. The following information, among other things, is included in the profile:

- Import templates that you have created using the Text Import Wizard.
- All adjustments that you have made to the FlexPro user interface.
- Settings that you have made in the Analysis Wizard.
- Custom models that you have created for non-linear curve fitting.
- User rights and administrator password.

You can export the user profile to a file and import it from a file to transfer the profile to another user.

### User Rights

In conjunction with the customization of the FlexPro user interface, user rights help to distinguish between developers who create analyses and documents using FlexPro, and users who use these ready-made analyses to evaluate their measurements. You can restrict user rights to protect specific analyses and documentation created using FlexPro against incorrect operation and manipulation. User rights are assigned to the user logged into Windows and are stored in the user's profile.

You can protect the settings for user rights from unauthorized changes by setting an administrator password.

---

**Note:** User profiles and user rights cannot be edited using FlexPro View.

---

## Working with User Profiles and User Rights

### Assigning User Rights

1. Click on **Options** from the **Tools** menu.
2. In the **Options** dialog box, click on the **User Profile** tab.
3. Select the permissions that you or the user receiving a user profile is to have.

**Note:**

- If you have already created an administrator password, you must authenticate yourself in order to change user rights.
  - Some user rights cannot be enabled in the runtime version of FlexPro.
- 

**Creating a User Profile for Another User**

1. First save your own user profile:
    1. In the **Tools** menu, click on **Options** and then click on the **User Profile** tab.
    2. Click on **Export** to export your user profile to a file.
    3. If necessary, select a different folder and click **Save**.
  2. Customize the FlexPro user interface the way that you want the other user to see it. You can find information on how to do this under Customizing the User Interface.
  3. Now adjust the settings for the user's rights:
    1. In the **Tools** menu, click on **Options** and then click on the **User Profile** tab.
    2. Select the user rights that you want to assign to the user.
    3. To protect user rights, click on **Change Administrator Password** and enter the same password in the **New password** and **Confirm password** fields.
- 

**Note:** The password is stored in the user profile and from now on must be entered before any changes can be made to the user rights.

---

4. You can now save the user profile to a file:
  1. Click on **Export**.
  2. If necessary, select another folder.
  3. Enter a file name, for instance, the name of the user or the name of the application to be run using this user profile.
  4. Click on **Save**.
5. Now restore your own user profile:
  1. Click on **Import** to import your user profile from a file.
  2. Select the file you saved in the first step.
  3. Click on **Open**.

---

**Note:** User profiles cannot be created using FlexPro View.

---

### Installing a User Profile for Another User

To install profile that is stored in a file for another user, the following two methods can be used:

#### 1. Installation using Windows Explorer

a) Windows 7, Windows 8, Windows Vista

Copy the user profile to the user computer using the file name `UserProfile.fpup` In the folder `C:\Users\User login name\AppData\Local\Weisang\FlexPro 10` or `C:\>User>User login name>AppData>Local>Weisang>FlexPro>10`.

---

**Note:** If on the **System Settings** tab of the FlexPro **Options** dialog box you selected the option **Save user profile and personal template database on server**, then you will have to replace the folder `Local` with `Roaming` in the paths mentioned above.

---

b) Windows 2000, XP

Copy the user profile to the user computer using the file name `UserProfile.fpf` In the folder `C:\Documents and Settings\User login name\Local Settings\Application Data\Weisang\FlexPro 10`.

---

#### Notes:

The folders are hidden. If you cannot see them in Windows Explorer, you will need to go to the **Folder Options** dialog box under **View** and then select the option **Show hidden files and folders**.

FlexPro should not be running on your computer while you are installing the profile.

---

#### 2. Installation using FlexPro

1. Log into the user's computer using the user's login name.
2. Launch FlexPro.
3. In the **Tools** menu click **Options** and then click on the **User Profile** tab.
4. Click **Import**.
5. Select the file and click **Open**.

**Saving a User Profile On a Server (Roaming)**

Do the following if you want to copy your user profile to all computers you log into on the LAN:

- 1. Select **Options** from the FlexPro **Tools** menu.
- 2. In the **Options** dialog box, click on the **System Settings** button.
- 3. Select the option **Save user profile and personal template database on server**.

---

**Note:** Your computer must be a member of a Windows domain and your Windows roaming user profile must be enabled.

---

**7.3 Using Startup Switches to Launch FlexPro**

- 1. Do one of the following:  
If you use Windows Start menu to launch FlexPro, use Windows Explorer to locate the FlexPro shortcut icon in the Start Menu folder of your Windows folder. Right-click on the shortcut icon, click on **Properties** and then click on the **Shortcut** tab.
- 2. Under **Target**, type a space after the path to the FlexPro program, and then type one or more of the parameters listed in the following table.  
For example, to open the project database Measurement.fpd, type a space after the path and then type: C:\Project Databases\Measurement.fpd

To Start FlexPro and...	Enter
open a specific project database using a backup copy	project database path\file name or <b>/om:b</b> project database path\file name
open a specific project database directly	<b>/om:d</b> project database path\file name
open a specific project database as read-only	<b>/om:r</b> project database path\file name
prevent display of the FlexPro startup screen and a new blank project database	<b>/e</b>
specify the working folder	<b>/path</b> folder path\folder name

---

**Note:** The folder specified in the startup switch **/path** is used instead of the location specified in the **Default working folder** on the **System Settings** tab (**Tools** menu, **Options** command).

---



## 7.4 Customizing the FlexPro Undo Function

1. In the **Tools** menu, click on **Options**.
2. In the **Options** dialog box, click on the **Operation** tab.
3. Select the relevant options.

## 7.5 Customizing FlexPro Messages

You can turn off some of the FlexPro messages to avoid unnecessary repetitions. FlexPro will then display the option **Don't ask this question again** under the message. You can select this option to turn off the message.

To re-activate these messages, do the following:

1. In the **Tools** menu, click on **Options**.
2. In the **Options** dialog box, click on the **Messages** tab.
3. Select the relevant options.

## 7.6 Customizing the Unit for Dimension Measures

FlexPro can display dimension measures, such as the size of page margins, in different units. To select this unit, you proceed as follows:

1. In the **Tools** menu, click on **Options**.
2. In the **Options** dialog box, click on the **Regional Settings** tab.
3. Select the desired unit under **Unit of measure**.

## 7.7 Customizing the Decimal Symbol

You can customize the decimal symbol used for the input and output of floating point values as follows:

1. In the **Tools** menu, click on **Options**.
2. In the **Options** dialog box, click on the **Regional Settings** tab.
3. Select the appropriate mode from the **Decimal symbol** list box.

---

**Note:** This setting does not affect the decimal symbol in formulas. FPScript always uses the decimal point.

---

## 7.8 Adjusting the Output Format for Numeric Data

1. Select **Properties** from the FlexPro **File** menu.
2. In the project database Properties dialog box, click on the **General** tab.

**To set the number of valid decimal places when outputting floating point values:**

- Under **Maximum number of valid places for output of numeric data** enter the desired number of significant digits.

**To set the output format for complex numbers:**

- In the **Display complex numbers** list box, select **with imaginary unit** if the complex numbers are to be output in the format  $a+bi$ , or select **separated with separator** if they are to be displayed in the format  $(a, b)$ .

---

**Note:** FPScript accepts both notations for complex numbers. The comma must be always used as the list separator, however.

---

## 7.9 Application Profile

By authorizing Weisang to record and transmit a user profile for FlexPro, you are assisting the Weisang developer team in its further development.

The application profile contains information about how you use the software, i.e. which commands you use and how often you use them, the help topics you view, and dialog boxes you open. It also provides information on the operating system you are using and the system settings you use to run FlexPro.

The application profile does not contain any personal or confidential information such as your name, keyboard input, passwords or hard drive data.

When launching FlexPro for the first time, you can specify whether you authorize recording of the application profile. FlexPro then transmits the data every 30 days to Weisang. If you are interested in the recorded profile, FlexPro will show it to you before

it is transmitted.

You can change your settings for recording and transmitting the application profile at any time in the **Options** dialog box on the **User Profile** tab. To do this, select **Options** from the **Tools** menu.

## 8 Automating Processes

### 8.1 Creating and Using Macros

#### Using Macros to Automate Tasks

##### What Is a Macro?

If you want to perform a task repeatedly in FlexPro, you can automate the task by using a macro. A macro is a series of FlexPro commands and statements that you group together as a single command to accomplish a task automatically. Instead of manually performing a series of time-consuming, repetitive actions in FlexPro, you can create and run a single macro - in effect, a custom command - that accomplishes the task for you.

Here are some typical uses for macros:

- To speed up routine editing and formatting
- To combine several commands, such as importing data and creating a diagram
- To make an option in a dialog box more accessible
- To automate a complex series of tasks

FlexPro offers two ways for you to create a macro: the macro recorder and the Visual Basic Editor. The Visual Basic Editor, however, is only available in FlexPro Professional and FlexPro Developer Suite.

##### Recording a Macro

The macro recorder is the most convenient way of creating a macro. Even if you are working with FlexPro Professional or FlexPro Developer Suite and have the Visual Basic Editor available for programming, it makes sense to start with a macro recording and to modify the code later, if necessary. FlexPro records a macro as a series of FlexPro commands in Visual Basic for Applications. When you record a macro, you can use the mouse to select commands and options. You can temporarily pause recording and then resume recording where you stopped.

To simplify repetitive processes, FlexPro offers the Quick Macro. You can easily start and stop recording this macro using the CTRL+SHIFT+R shortcut. These macros do not have to be named or created. When you start a new recording, an existing Quick Macro is automatically deleted. You can run the Quick Macro as often as you like by using the CTRL+SHIFT+P shortcut.

For quick access to your macro, you can assign it to a toolbar, a menu, or shortcut keys. Running the macro is as simple as clicking on the toolbar button or menu command or pressing the shortcut keys. You can also use the **Tools** menu, select **Macros** from the

**Macro** submenu) and click on the name of the macro to be executed.

### **Creating a Macro from Scratch**

You can use the Visual Basic Editor of FlexPro Professional or FlexPro Developer Suite to create very flexible, powerful macros that include Visual Basic instructions that you cannot record.

### **Where Macros Are Stored**

You can store macros in templates databases or in project databases. By default, FlexPro stores macros in the personal templates database so that they are available for use with every FlexPro project database. If you plan to use a macro for only one particular project database, store it in that project database. Individual macros in a project database are stored in macro projects that you can copy from one project database to another.

## **Protection from Documents That May Contain Viruses**

### **Understanding Macro Viruses**

A macro virus is a computer virus stored in a Macro inside a project database, a template database or an add-in. If you open this type of project database or start a procedure that calls a macro virus, the macro virus can become enabled, can be transferred to your computer and be stored in your personal template database. From that point on, every project database you open could be automatically "infected" with the macro virus - and if other people open these infected project databases, the macro virus is transmitted to their computers.

### **Security Levels in FlexPro**

FlexPro offers the following levels of security to reduce the chances that macro viruses will infect your project databases, templates databases or add-ins:

- **High** You can run only macros that have been digitally signed and that you confirm are from a trusted publisher. Before trusting a source, you should confirm that the source is reliable and uses a virus scanner before signing macros. Unsigned macros are automatically disabled, and FlexPro opens the project database without any warning.
- **Medium** FlexPro displays a warning whenever it encounters a macro from a source that is not on your list of trusted publishers (described further down). You can choose whether to enable or disable the macros when you open the project database. If the project database might contain a virus, you should choose to disable macros.

- **Low** If you are sure that all the project databases and add-ins you open are safe, you can select this option, which turns off macro virus protection in FlexPro. At this security level, macros are always enabled when you open project databases.

If your network administrator has not enforced a security level for your organization, you can change the security level. If the security level for FlexPro is set to Medium or High, you can maintain a list of trusted macro sources. When you open a project database or load an add-in that contains macros developed by any of these sources, the macros are automatically enabled. Learn about Security Levels in FlexPro.

### Digital Signatures

If you have Microsoft Internet Explorer version 4.0 or later installed on your computer, one way to identify that a macro is safe to use is by its digital signature. A digital signature on a macro is like a stamp on an envelope - it confirms that the macro originated from the macro developer who signed it and that the macro has not been altered.

When you open a project database or load an add-in that contains a digitally signed macro, the digital signature appears on your computer as a certificate. The certificate names the macro's source, plus additional information about the identity and integrity of that source. A digital signature does not necessarily guarantee the safety of a macro, and you must decide whether you trust a macro that has been digitally signed. For example, you might trust macros signed by someone you know or by a well-established company. If you are unsure about a project database or add-in that contains digitally signed macros, carefully examine the certificate before enabling macros or, for the highest level of security, disable the macros. If you know you can always trust macros from a particular source, you can add that macro developer to the list of trusted publishers when you open the project database or load the add-in.

If you yourself develop macros with FlexPro Professional or FlexPro Developer Suite, you can sign macros from within the Visual Basic Editor.

### Digital Signatures

#### How Digital Signatures Work

FlexPro has security levels that allow users to run macros based on whether or not they have been digitally signed by a macro developer on their list of trusted publishers. FlexPro also checks whether the digital signature is valid. For example, FlexPro checks whether the Macro Project was changed or signed before the digital certificate expired or before it was declared invalid. In addition, it checks whether the digital certificate was distributed by a valid certification authority.

Digital signatures work only on computers that have Microsoft Internet Explorer version 4.0 or later installed. If a user tries to open a project database that contains macros (signed or not) on a computer that does not have Internet Explorer 4.0 or later installed, a standard macro virus dialog box appears giving the user options for enabling or disabling macros before the project database is opened.

## **Signing Macro Projects**

**How to proceed** FlexPro uses Microsoft Authenticode technology to enable you to digitally sign macro projects you develop. First, you must obtain a digital certificate and install it. Then, test your solution - whenever code in a signed macro project is modified in any way, its digital signature is removed. When your solution is ready for distribution, sign the macro project. If you have the proper digital certificate on your computer, the macro project will automatically be re-signed when you save it. If you want to prevent users of your solution from accidentally modifying your macro project and invalidating your signature, lock the macro project before you sign it.

**What a digital signature does not do** Your digital signature says only that you guarantee that this project is safe. It does not prove that you wrote the macro project. Locking and signing your macro project does not prevent another macro developer from replacing the digital signature with another signature. For example, corporate administrators may re-sign templates and add-ins so that they can control exactly what users can run on their computers.

**Add-ins and digital signatures** If you create an add-in that adds code to a macro project, it is recommended that your code checks whether the project is digitally signed. Users should be notified of the consequences of modifying a signed project before they continue. Modifying a project database that contains a macro project does not invalidate the signature. It is not the project database that is signed, but the macro project.

## **Adding a Macro Developer to the List of Trusted Publishers**

To use this procedure, you must have Microsoft Internet Explorer version 4.0 or later installed on your computer and you must set the Security Level to "Medium" or "High".

1. Open the project database or load the add-in that contains macros from the source you want to add to the list.
2. In the **Security Warning** field, check the box **Always trust macros from this source**.

---

**Note:** If the box **Always trust macros from this source** in the **Security Warning** field is not available, then the macros were not digitally signed. You cannot add this macro developer to the list of trusted publishers without a digital signature.

---

## Changing the Security Level for Macro Virus Protection

1. In the **Tools** menu, click on **Macro** and then click on **Security**.
2. Click on the **Security Level** tab and then choose the preferred security level.

## Creating a Macro

You can create a macro by using the macro recorder to record a sequence of actions. With FlexPro Professional or FlexPro Developer Suite, you can create a macro from scratch by entering Visual Basic for Applications code in the Visual Basic Editor. When you are recording a macro, you can use the mouse to click commands and options.

### To Record a Macro in FlexPro

1. In the **Tools** menu, point to **Macro** and then click on **Record**.
2. In the **Macro name** field, enter the macro name.
3. Under **Store macro in**, enter the templates database or the project database where you want to store the macro.
4. Under **Description** enter a description for the macro.
5. Click on **OK** to start recording the macro.
6. Perform the actions you want to include in your macro.
7. To stop recording the macro, click on **Stop recording**.

### To Record a Quick Macro in FlexPro

1. In the **Tools** menu, click on **Record Quick Macro** or use the CTRL+SHIFT+R shortcut.
2. Perform the actions you want to include in your macro.
3. To stop recording your macro, click on **Stop Recording** or use the CTRL+SHIFT+R shortcut again.



### To Pause and Resume a Macro Recording

1. To pause recording, click on **Pause recording**.
2. Perform any actions you don't want to record.
3. To resume recording, click on **Resume recording**.

### To Create a Macro from Scratch

1. On the **Tools** menu, point to **Macro** and then click on **Macros**.
2. In the **Macros in** list, click on the templates database or the database where you want to store the macro.
3. In the **Macro name** field, enter the macro name.
4. Click on **Create** to open the Visual Basic Editor.

### Tips for Recording a Macro

- Before you record or write a macro, plan the steps and commands that you want the macro to perform. If you make a mistake when you record the macro, corrections you make will also be recorded. If you are working with FlexPro Professional or FlexPro Developer Suite, you can edit the macro later and remove unnecessary steps you recorded.
- Try to anticipate what is going to happen. For example, if your macro includes a command to close the project database, FlexPro asks you to save the project database if it contains unsaved changes. To avoid this message, record saving the project database before closing it.
- If you want to use the macro you are recording in other project databases, make sure that the macro does not depend on the current project database's contents.
- If you use a particular macro often, assign it to a toolbar button, a menu, or shortcut keys. This allows you to run the macro directly without having to open the **Macros** dialog box.

### Editing a Macro

FlexPro Professional and FlexPro Developer Suite allow you to edit your macros. Open a macro in the Visual Basic Editor and make corrections, remove unnecessary steps, rename or copy individual macros, or add instructions that you cannot record in FlexPro. Changes you make in the Visual Basic Editor to procedures and macro project items are

reflected in FlexPro in the **Macros** and **Organize Macro Project Items** dialog boxes.

1. In the **Tools** menu, point to **Macro** and then click on **Macros**.
2. Under **Macro name**.

If the macro is not displayed in the list, select a different project database or a different template database under **Macros**.

3. Click on **Edit**.

## Running a Macro

### To Run a Named Macro

1. In the **Tools** menu, point to **Macro** and then click on **Macros**.
2. Under **Macro name**, click on the name of the macro you want to delete.
3. If the macro is not displayed in the list, select a different project database under **Macros in**.
4. Click on **Run**.

### To Run the Quick Macro

- In the **Tools** menu, click on **Run Quick Macro** or use the CTRL+SHIFT+P shortcut.

## Deleting a Macro

1. In the **Tools** menu, point to **Macro** and then click on **Macros**.
2. Under **Macro name**, click on the name of the macro you want to delete. If the macro is not displayed in the list, select a different project database or a different template database under **Macros in**.
3. Click on **Delete**.

---

**Note:** To delete several macros, hold down the CTRL key and click on the macros you want to delete in the **Macro name** box and then click on **Delete**.

---

## Copying a Macro Project

You can use the **Organize Macro Project Items** dialog box to copy a macro project and use it in another project database or another database template.

1. In the **Tools** menu, click on **Macro** and then click on **Macros**.
2. Click on **Organize**.
3. Select the macro project you want to copy from one of the lists, and then click on **Copy**.

FlexPro displays the macro projects used in the active project database in the list on the left and the macro projects in the personal templates database in the list on the right.

---

**Note:** To copy an individual macro, select the macro in the **Macros** dialog box (**Tools** menu, **Macro** submenu), click on **Edit**, and use the standard editing features of the Visual Basic Editor. (Requires FlexPro Professional or FlexPro Developer Suite)

---

## Renaming a Macro Project

You can use the **Organize Macro Project Items** dialog box to rename a macro project. You can also rename an individual macro..

1. In the **Tools** menu, point to **Macro** and then click on **Macros**.
2. Click on **Organize**.
3. Select the macro project you want to rename from the respective list, and then click on **Rename**.
4. The macro projects in the currently active project database are displayed in the list on the left and the macro projects in the personal templates database in the list on the right.
5. Specifies a new name for the macro project under **New name**.

## Deleting a Macro Project

You can use the **Organize Macro Project Items** dialog box to copy a macro project. You can also delete an individual macro.

1. In the **Tools** menu, point to **Macro** and then click on **Macros**.
2. Click on **Organize**.
3. Select the macro project you want to delete from the appropriate list, and then click

on **Delete**.

4. FlexPro displays the macro projects used in the active project database in the list on the left and the macro projects in the personal templates database in the list on the right.

## Digitally Signing a Macro Project

For additional information on how macro virus protection works, you can learn about digital signatures and how security levels and digital signatures work together.

1. Open the project database or template that contains the macro project you want to sign.
2. In the **Tools** menu, point to **Macro** and then click on **Visual Basic Editor**.
3. In the Project Explorer, select the project that you want to sign.
4. In the **Tools** menu, point to **Digital Signature**.
5. Do one of the following:
  - To use the current certificate, click **OK**.
  - If you did not previously select a digital certificate, or if you want to use different one, click **Choose**, select the certificate, and then click **OK**.

---

**Note:** This function is only available in FlexPro Professional and FlexPro Developer Suite.

---

## Obtaining a Digital Certificate

### Creating your own digital certificate

You can create your own digital certificate by installing and using the digital certificate program that came with FlexPro Professional and FlexPro Developer Suite.

- In Windows Explorer, go to C:\Program Files (or Program Files (x86))\Shared Documents (or Common Files)\Microsoft Shared\OFFICE11 and double-click on the file called SelfCert.exe.

**Note:** Since a digital certificate that you create yourself is not issued by a formal certification authority, macro projects that you digitally sign by using such a certificate are referred to as self-signed projects. Depending on how FlexPro digital signature features are handled in your organization, you may be prevented from using this type of certificate, and other users might not be able to run self-signed macros for security reasons.

---

### **Request a digital certificate from your organization**

Some organizations and corporations may have a security administrator or group act as their own certification authority and produce or distribute digital certificates by using tools such as Microsoft Certificate Server. Microsoft Certificate Server can act as a stand-alone certification authority or as part of an existing certification authority hierarchy. This depends on how the FlexPro features for digital signatures are used in your organization. You can sign a macro project by using a digital certificate from your organization's internal certification authority. Or you might need to have an administrator sign macro projects for you by using an approved certificate. For information about your organization's policy, contact your network administrator.

### **Request a digital certificate from a commercial certification authority**

To obtain a digital certificate from a commercial certification authority (such as VeriSign, Inc.), you or your organization must submit an application to that authority.

Depending on your status as a macro developer, you should apply for either a Class 2 or Class 3 digital certificate for software publishers:

- A Class 2 digital certificate is designed for people who publish software as individuals. This class of digital certificate provides assurance as to the identity of the individual publisher.
- A Class 3 digital certificate is designed for companies and other organizations that publish software. This class of digital certificate provides greater assurance about the identity of the publishing organization. Class 3 digital certificates are designed to represent the level of assurance provided today through retail distribution channels for software.

When you receive your digital certificate, you are given instructions on how to install it and use it to sign your FlexPro solutions.

## Removing a Macro Developer from the List of Trusted Publishers

To perform these procedures, you must have Microsoft Internet Explorer version 4.0 or later installed on your computer.

1. In the **Tools** menu, point to **Macro** and then click on **Security**.
2. Click on the **Trusted Publishers** tab.
3. Select the source you want to remove from the list.
4. Click on **Remove**.

## Troubleshooting Macro Warnings and Security Levels

### Receiving Macro Warnings

No Warning Displayed When Opening a Project Database or Loading an Add-In Containing Macros.

#### **The security level for FlexPro might be set to Low**

To have FlexPro warn you that a database or add-in contains macros, change the security level to 'Medium'.

#### **The security level for FlexPro might be set to 'High', and the macros are not digitally signed**

To have FlexPro warn you that a database or add-in contains macros that are not digitally signed, change the security level to 'Medium'.

#### **You or someone you have designated as a trusted publisher might have developed the macros**

If you designated the macro developer as a trusted publisher, FlexPro opens the database and enables macros. If you no longer consider the macro developer a trusted source, you can remove the macro developer from the list of trusted publishers. To have FlexPro prompt you every time you open a database or load an add-in that contains macros, remove every publisher from the list of trusted publishers. If you are using FlexPro Professional or FlexPro Developer Suite, you can check whether the macros contained in the database have been signed. On the **Tools** menu, point to **Macro** and then click on **Visual Basic Editor**. In the Visual Basic Editor, click on **Digital Signature** in the **Tools** menu.

A Macro Warning Keeps Appearing.

#### **The macro you want to run might not be from a trusted publisher**

You will receive a macro warning if the security level for FlexPro is set to Medium or

High, and you open a database or load an add-in that contains digitally signed macros that are not from a trusted publisher. If you are sure that the macro developer is a trusted publisher, add the name to the list of trusted publishers.

**The database may contain a macro virus**

If you do not expect the database to contain a macro, your computer might have a virus that is adding a macro virus to the database. Check your computer for viruses, and try to get an uninfected copy of the database from the source.

**The database contains legitimate but unsigned macros**

FlexPro cannot distinguish between safe macros and unsafe macros. If you know the macros are legitimate and safe, you might want to digitally sign those macros and then add their names to the list of trusted publishers.

**FlexPro Should Not Display Macro Warnings.**

To stop displaying macro warnings, change the security level for FlexPro to Low. To prevent your computer from becoming infected by macro viruses, you should only set your security level to **Low** if you use a separate anti-virus application that can check the FlexPro databases and add-ins for macro viruses. Otherwise, you should make sure that all macros used by you come from trusted publishers.

**A Security Warning Dialog Box Appears Stating That the Source has not been Authenticated.**

This warning appears in the **Security Warning** box if the security level for FlexPro is set to High or Medium, and you open a database or load an add-in that contains digitally signed macros, but the digital certificate has not been authenticated. For example, if the macro developer has created his or her own digital certificate, you will receive this warning. This type of unauthenticated certificate can be forged by malicious users to falsify the identity of the source of the certificate. For example, a malicious user could create a certificate with the description "Weisang & Co." The only warning that the certificate is false would come from the warning described here. You should expect professional software developers to sign using authenticated certificates. You should only accept unauthenticated certificates from co-workers you know or friends. Don't accept any from sources you don't know.

If the security level for FlexPro is set to High and you trust the macro source, you can select the **Always trust macros from this source** check box and enable the macros. If the security level for FlexPro is set to Medium, you can enable the macros without adding the macro developer to the list of trusted publishers. If you don't consider this macro developer a trusted publisher, do not enable the macros or do not open the database until you have verified that the publisher can be trusted.

## Working with Macros

Cannot Use a Macro in an Open Project Database or Add-In.

### Macros might have been automatically disabled

If the security level for FlexPro is set to High and you open a database or load an add-in that contains unsigned macros, the macros are disabled and you cannot run them. You can enable macros that are not digitally signed by changing the security level to **Medium**, closing the data base or unloading the add-in, and then opening the database or loading the add-in again. Remember to change the security level back to **High** if you want FlexPro to automatically disable unsigned macros in the future.

### You might have chosen to disable macros when you opened the database

If the security level for FlexPro is set to **Medium** or **High**, and you have determined that the macros are to be disabled because their source is not trustworthy, you cannot run the macros. To run the macros, close the database or unload the add-in, re-open the database or load the add-in and then click **Enable Macros**.

## Changing the Security Level

Cannot Change the Security Level for FlexPro.

Your network administrator might have enforced a security level for your workgroup or corporation to make sure you use only macros that have been deemed virus free. For more information, contact your network administrator.

## Adding a Macro Developer to the List of Trusted Publishers

Cannot Add a Specific Macro Developer to the List of Trusted Publishers.

### The list of trusted publishers might be locked

Your system administrator may have locked your list of trusted publishers so that you cannot add new macro developers to it.

### The macro might not be digitally signed

In order to add a macro developer to the list of trusted publishers, the macro must be digitally signed by the macro developer.

### Microsoft Internet Explorer version 4.0 or later may not be installed on your computer

Internet Explorer 4.0 or later is required in order to digitally sign a macro or verify that a macro developer is a trusted publisher.



**The security level for FlexPro might be set to Low**

In order to add a macro developer to the list of trusted publishers, you must set the security level to 'Medium' or 'High'.

The Source of the Certificate I Chose to Trust Was Not Added to the List of Trusted Sources.

If you use Microsoft Internet Explorer version 4.0 with FlexPro, the **Certificate Properties** dialog box appears when you click the **Details** button in the **Security Warning** dialog box to get information about a digital certificate for a digitally signed macro in a FlexPro database.

The **Trust** tab in the **Certificate Properties** dialog box contains an option for categorizing the certificate as generally trustworthy. FlexPro, however, ignores all options that you have activated on the **Trust** tab. To add a macro developer to the list of trusted publishers when you load databases, check the box **Always trust macros from this source** in the **Security Warning** dialog box.

**Troubleshooting When Recording and Running Macros**

While recording a macro an unwanted action was inadvertently recorded.

If you are using FlexPro Professional or FlexPro Developer Suite and want to undo an unwanted action in a macro, you can open the macro in the Visual Basic Editor and remove any unwanted steps. How to edit a macro.

If you don't want to use the Visual Basic Editor, or it is not available to you, record the macro again without the unwanted action.

Recording a macro occasionally produces an error message.

A macro you record may not run properly in every situation. If the macro cannot run, FlexPro displays an error message. Some macros depend on certain options or settings in FlexPro. For example, a macro that edits a selected object won't run properly if no object is selected. If a macro you've recorded produces an error message, note the error number. You can then search for "error messages" in Help and find information about the message you received.

Cannot edit macros while project database is open.

If you are editing macros in a read-only database, you cannot save the changes you made. For a read-only database, FlexPro adds **[Read-Only]** to the file name in the title bar.

Macro cannot be recorded if project database is open.

Recorded macros cannot be saved in a read-only database. For a read-only database, FlexPro adds **[Read-Only]** to the file name in the title bar.

Cannot run macro while project database is open.

Changes resulting from running macros cannot be saved in a read-only database. For a read-only database, FlexPro adds **[Read-Only]** to the file name in the title bar.

## Security Levels in FlexPro

The following table summarizes the functionality of macro virus protection, taking into account the settings specified on the **Security Level** tab of the FlexPro **Security** dialog box (**Tools** menu, **Macro** submenu) and various conditions.

Condition	High	Medium	Low
Unsigned macros.	Macros are deactivated automatically and the database is opened.	Users are asked to activate or deactivate macros.	No message. Macros are activated.
Signed macros from a reliable source. Signature is valid.	Macros are activated automatically, and the database is opened.	Macros are activated automatically, and the database is opened.	No message or verification of signature. Macros are activated.
Signed macros from an unknown author. Signature is valid.	The program shows a dialog box with information on the certificate. Users can only activate macros if they regard the author and the certification authority as a reliable source. A network administrator can block the list of reliable sources and thus prevent users from adding the developer of the macro to the list and activating macros.	The program shows a dialog box with information on the certificate. Users are prompted to activate or deactivate macros. Users can decide whether to trust the developer of the macro and the certification authority.	No message or verification of signature. Macros are activated.
Signed macros from any author. The signature is invalid, possibly because of a virus.	Users are warned of a potential virus. Macros are deactivated automatically.	Users are warned of a potential virus. Macros are deactivated automatically.	No message or verification of signature. Macros are activated.
Signed macros from any author. Verification is not	User is warned that signature verification is not possible. Macros are	User is warned that signature verification is not possible. The user is	No message or verification of signature. Macros

possible because the public key is missing or incompatible encryption methods have been used.	deactivated automatically.	asked to activate or deactivate macros.	are activated.
Signed macros from any author. The macro project was signed after the certificate had expired or had been revoked.	User is warned that the signature has expired or has been revoked. Macros are deactivated automatically.	User is warned that the signature has expired or has been revoked. The user is asked to activate or deactivate macros.	No message or verification of signature. Macros are activated.

## 8.2 Automation Using FlexPro Visual Basic

### Getting Started with FlexPro Visual Basic

#### Objects, Understanding Properties and Methods

Objects are the fundamental building blocks of Visual Basic. Virtually everything you do in Visual Basic involves modification of objects. Every FlexPro element (databases, diagrams, tables, documents, data sets, etc.) can be represented by a Visual Basic object.

#### What are Objects and Collections?

An object represents a FlexPro element, such as a database, a diagram, or an individual curve in a diagram. A collection is an object containing several other objects that are usually of the same kind. All the curves in a diagram, for example, are contained in a single collection object. Using properties and methods, you can modify an individual object or a whole collection of objects.

#### What is a Property?

A property is an attribute of an object or an aspect of its behavior. For example, properties of a diagram include its name, the appearance of the division grid and whether or not the cursors are enabled. To change the characteristics of an object, you need to change the values of its properties.

To specify the value of a property, first reference the object, then enter a dot, the property name, an equals sign and the new value of the property. In the following example, a new comment is assigned to the root folder of the **MyDatabase** database.

```
Databases("MyDatabase.fpd").RootFolder.Comments = "My Database"
```

In this example, **Databases** references the collection of open databases. The name **MyDatabase.fpd** identifies a single database in the collection. The **Comments** property specifically applies to the root folder of this database.

Some properties cannot be set. The help topic for a property specifies whether this property can be set (read/write access) or whether it can only be read (read-only access).

You can obtain information about an object by calling the value of one of its properties so that it will be returned. In the following example, the name of the active database is returned.

```
databaseName = ActiveDatabase.Name
```

In this example, **ActiveDatabase** references the database in the active window in FlexPro. The name of this database is assigned to the variable **databaseName**.

---

**Note:** The help topic of a property specifies whether you can set this property (write-read access), only read it (read-only access) or only write to it (write-only access). The Object Browser in the Visual Basic-Editor also displays the read/write status at the bottom of the browser window when the property is selected.

---

### What is a Method?

A method is an action that can be executed by an object. For example, since a FlexPro document can be printed, the Document object also features a PrintOut method. Methods often have arguments that more clearly define the way the action is to be executed. In the following example, the first three pages of the active document are printed.

```
ActiveDatabase.ActiveObject.PrintOut FirstPage:=1, LastPage:=3
```

In most cases, methods are actions and properties are characteristics. The use of a method means that something happens to an object, whereas the use of a property returns information on the object or causes an attribute of the object to be changed.

### Returning an Object

Most objects are returned by returning an individual object from a collection. The **Databases** collection, for example, contains the open FlexPro databases. Use the **Databases** property of the **Application** object (the topmost object in the FlexPro object hierarchy) to return the **Databases** collection.

Once you have accessed the collection, you can return an individual object by specifying an index value in brackets (similar to working with arrays). The index value is usually a number or a name.

In the following example, the **Databases** property is used to access the **Databases** collection. The index number is used to return the first database in the **Databases** collection. The **Close** method is then applied to the **Database** object to close the first database in the **Databases** collection.

```
Databases(1).Close
```

In the following example, a name (specified as a string) is used to identify a **Database** object within the **Databases** collection.

```
Databases("Report.fpd").Close
```

Collection objects often have methods and properties which can be used to modify the complete collection of objects. The **Databases** object has a **Save** method that saves all documents in the collection. In the following example, the open documents are saved using the **Save** method.

```
Databases.Save
```

The **Database** object also features a **Save** method for saving an individual database. In the following example, the **Report.fpd** database is saved.

```
Databases("Report.fpd").Save
```

If an object located further down in the FlexPro object model is to be returned, you have to "drill down" to this object by using properties and methods that return objects.

To see how this works, open the Visual Basic Editor and in the **View** menu click on **Object Browser**. In the **Classes** list on the left, click on **Application**. In the members list on the right, click on **ActiveDatabase**. The text at the bottom of the Object Browser shows that **ActiveDatabase** is a read-only property that returns a **Database** object. At the bottom of the Object Browser, click on **Database**. The **Database** object is then automatically selected in the **Classes** list and the **Members** list shows the members of the **Database** object. Scroll through the list of members until you find **Close**. Click on the **Close** method. The text at the bottom of the **Object Browser** window shows the syntax for the method. For more information about this method, press **F1** or click on the **Help** button and navigate to the help topic covering the **Close method**.

Leveraging this information, you can write the following statement for closing the active database.

```
ActiveDatabase.Close SaveChanges:=fpSaveChanges
```

In the following example, the caption of the active window is changed.

```
ActiveDatabase.ActiveWindow.Caption = "My Window"
```

The **ActiveWindow** property returns a **Window** object that displays the active window. Use the **Caption** property to specify the title of the window.

In the following example, a new database is created and the **Save As** dialog box appears so that the database can be given a name.

Databases.Add.Save

The **Databases** property returns a **Databases** collection. The **Add** method creates a new database and returns a **Database** object. The **Save** method is then applied to the **Database** object.

As you can see, methods and properties are used to drill down to an object. This means that you return an object by applying a method to a higher-level object in the object hierarchy or accessing a property. Once you have returned the required object, you can use the methods of this object and control its properties. For an overview of the object hierarchy, go to FlexPro Object Model.

### How to Get Help on Objects, Methods and Properties

Until you are familiar with the FlexPro Object Model, you can use a few resources that will help you to drill down through the hierarchy.

- Listing Members automatically. Typing a period (.) after an object in the Visual Basic Editor displays a list of the available properties and methods. If, for example, you enter **Application.**, a drop-down list of the methods and properties of the **Application** object appears.
- Help. You can also use Help to find out which properties and methods can be used for an object. Every object topic in Help contains properties and methods sections, which list the properties and methods for the object. In the Object Browser or in a module, press **F1** to get to the corresponding help topic.
- FlexPro Object Model. This topic explains how FlexPro objects are arranged in the hierarchy. Click on an object in the image to display the corresponding help topic.
- Object Browser. The Object Browser in the Visual Basic Editor displays the members (properties and methods) of the FlexPro objects.

### Understanding Events

Events take place when certain program situations occur. These events can be handled in event procedures initiated by the user.

### Using Events

In FlexPro, you can implement event procedures at the database, object class (AnyCursorObject, AnyDocObject, AnyFolder, AnyFormula, AnyFpObject, AnyValueObject), object or application level.

The **ObjectOpened** event, for example, occurs at the object and object class level, whereas the **DatabaseSaved** event is available at the database and the application

level. The **DatabaseSaved** event for a database occurs after this database has been saved. At the application level, the **DatabaseSaved** event occurs once one of the open databases has been saved.

## Enabling Events

Using the `EnableEvents` property of the Application object, you can enable or disable the events.

For example, saving a database with the `Save` method triggers the `BeforeDatabaseSave` event. This can be prevented by setting the **EnableEvents** property to **False** before you call up the **Save** method.

```
Application.EnableEvents = False
```

```
ActiveDatabase.Save
```

```
Application.EnableEvents = True
```

By default, events in FlexPro are disabled, i.e. **Application.EnableEvents** is set to **False**. You can enable events by setting the **EnableEvents** property to **True** or in the **Tools** menu by changing the default settings under **Option | Default Settings** and checking the **Enable Macro Events** box.

---

**Note:** If you are creating an event procedure in an object module for the first time, you will be notified if the events option has been disabled.

---

## Event Types

There are basically two types of events in FlexPro. First, there are events that occur once a certain condition occurs. In this case, the application developer is only notified of the change of condition (e.g. `ValueModified`). The second type includes those events that can be influenced by the application developer. These events form sequences of up to three events. The sequence appears as follows: It starts with a **QueryCancel EventName** event, which allows the application developer to decide whether a certain program condition may occur. If this event returns **True**, FlexPro discontinues processing, triggering the **EventNameCancelled** event. If the **QueryCancel EventName** event is not processed or if it returns **False**, FlexPro will continue processing, triggering the **Before** event just before the event occurs. Eventually, once the program condition has been reached, the **EventName** event will be initiated.

The following VBA code describes this logic using the `ObjectOpened` event as an example:

```
' The oObject object is opened by double-clicking in the  
' list view.
```

```
If QueryCancelOpenObject(Object) Then
    OpenObjectCanceled oObject
Else ...
    BeforeOpenObject oObject
    oObject ... ' Object is opened ...
    ObjectOpened oObject
End If
```

### Forwarding Events

In FlexPro, events initiated for an object are not just signaled with this object, they are also forwarded to the template databases and/or parent objects in the object model. Thus, events of similar objects in the templates databases or in parent objects of the project database can be processed simultaneously.

For example, once the database "Data.FPD" has been saved, the following event procedures are executed one after the other, depending on whether or not they exist:

- **DatabaseSaved** event procedure of **ThisDatabase** of the "Data.FPD" database
- **DatabaseSaved** event procedure of **ThisDatabase** of the personal template database
- **DatabaseSaved** event procedure of the **Databases** collection of the "Data.FPD" database
- **DatabaseSaved** event procedure of the **Application** object

The first argument of an event is always the object concerned. Therefore, an event procedure to which the event was forwarded will know for which object the event just processed has occurred.

The object documentation in the Reference section will tell you to which objects events are forwarded.

### Forwarding Events for FlexPro Objects

Forwarding events is particularly important for FlexPro objects. Since a FlexPro database may include many FlexPro objects for which there is no corresponding object module in the Visual Basic Project Explorer, the best practice is to complete the event procedures for one, several or all FlexPro objects in the class object modules **AnyCursorObject**, **AnyDocObject**, **AnyFolder**, **AnyFormula**, **AnyFpObject** and **AnyValueObject**.

For example, if the **ObjectModified** event of the **FpObject** object occurs, the following event procedures are executed one after the other, depending on whether or not they



exist:

- **ObjectModified** event procedure of the object concerned
- **ObjectModified** event procedure of **AnyFpObject** in the database containing the object
- **ObjectModified** event procedure of **AnyFpObject** in the personal template database

The described event cascading applies to all FlexPro objects.

### **Tips on Working with FlexPro VBA**

This topic provides some information on issues that might occur when programming in Visual Basic.

### **Using the Set Statement When Assigning the Value Property to Variables**

When assigning the Value property of a ValueObjectobject to a variable, you have to consider that the Value property is of the Variant type and can also return complex values, signals and lists in addition scalar values and arrays. Complex values, signals and lists are themselves objects in FlexPro and have to be assigned to variables using the Set statement.

The following statement only represents a valid assignment of a value if the value of oValueObject is neither complex nor a signal or list:

```
Dim vtValue As Variant
Dim oValueObject As ValueObject

...

vtValue = oValueObject.Value
```

If **oValueObject.Value** is complex or a signal or list, the error message 'Object does not support this property or method' appears. When assigning complex values, signals, or lists, you should therefore use the **Set** statement:

```
set vtValue = oValueObject.Value
```

When you are writing a macro and you do not know whether a value will be complex or real at runtime, using the **IsObject** statement can avoid errors when the macro is run:

```
If IsObject(oValueObject.Value) Then
    Set vtValue = oValueObject.Value
Else
    vtValue = oValueObject.Value
```

End If

If the value of **oValueObject** is an object, the **Set** statement is used; otherwise, the value has to be assigned using '='.

### Problems when Using OLE\_COLOR Color Values

The **OLE\_COLOR** type is an OLE standard data type for storing RGB color values. In C/C++, the data type is declared as an unsigned Long value. VBA does not support unsigned values. Consequently, use of **OLE\_COLOR** may occasionally cause problems. The following code might generate the error message 'Variable uses Automation type not supported in Visual Basic':

```
if ActiveDatabase.ActiveObject.Selection.FillFormat.Color <> fpColorRed Then
    ...
End If
```

There are two ways to avoid this error message. The first is to convert the color values to a **Long** value before the comparison:

```
If CLng(ActiveDatabase.ActiveObject.Selection.FillFormat.Color) <> fpColorRed Then
    ...
End If
```

The second is to introduce an **OLE\_COLOR** variable, assign the color value to it, and then use the new variable for the comparison:

```
Dim clr As OLE_COLOR
clr = ActiveDatabase.ActiveObject.Selection.FillFormat.Color
If clr <> fpColorRed Then
    ...
End If
```

### Deleting FlexPro Objects from a Collection within a For...Each Loop

Deleting FlexPro objects from a collection using a **For...Each** loop should be avoided, since it could cause unwanted results, and some elements in the collection might be missed.

If the elements of any collection of FlexPro objects are to be deleted, the following code leads to the desired result:

```
While oColl.Count > 0
    oColl.Remove 1
```

```
Wend
Or
While oColl.Count > 0
    oColl(1).Delete
Wend
```

### Converting Numbers to Strings Using the CStr Function

When using the **CStr** function to convert numbers to strings, you should consider that the function always uses the current Region and Language Options settings (Locale) for the conversion. On a German system, a ',' is used to separate the integral part of a decimal number from its fractional part, whereas the '.' is used on a British or American system. If you want to convert numbers independently of the system settings, you have to use the Windows API function **SetLocaleInfo** to change the system settings temporarily. In the A Custom Import Filter example you can see how this API function is used.

### Preventing Error Messages for Fully Automated Processes

For fully automated processes that are to run without user interaction, you first have to make sure that all errors are handled by the macros themselves (consistent use of the **On Error** statement). Furthermore, make sure the following setting is enabled in the Visual Basic environment: under Tools | Options on the **General** tab, the setting for **Error Trapping** must be set to **Break on Unhandled Errors**. Otherwise, the execution of the program might be interrupted if allowed errors occur, and the error might be displayed, requiring user interaction (also see the DisplayAlerts property of the Application object).

### Evaluation of Expressions in VBA (Note for C/C++ Developers)

As opposed to C/C++, where expressions are only evaluated from left to right for as long as necessary during condition evaluation, VBA always evaluates the complete expression.

The following C/C++ expression

```
(1) if(exp1 || exp2)
    ...
```

corresponds to the following in VBA code

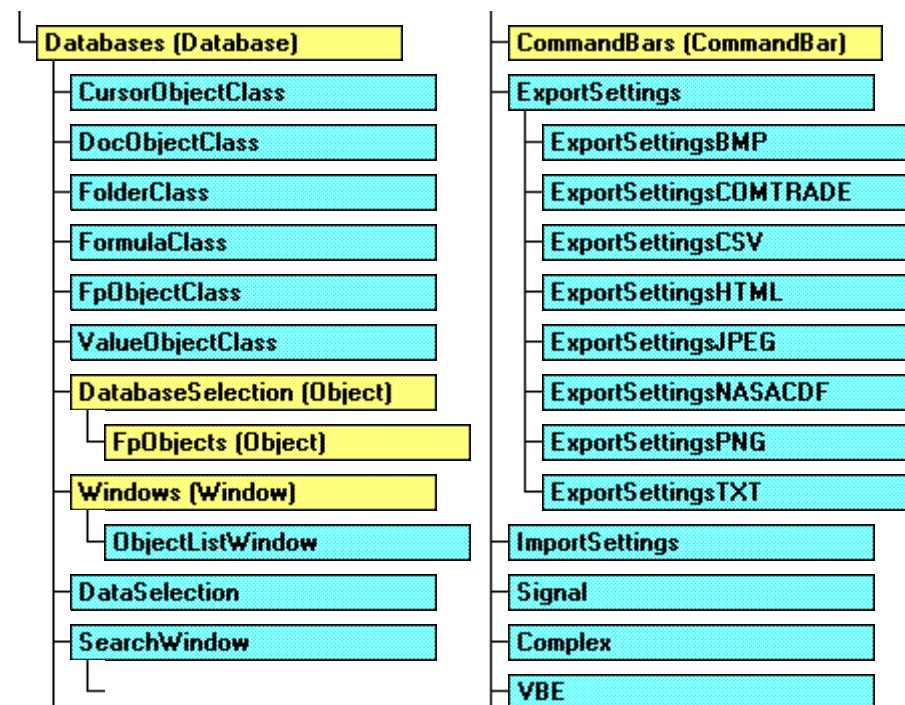
```
(2) If exp1 Or exp2 Then
    ...
```

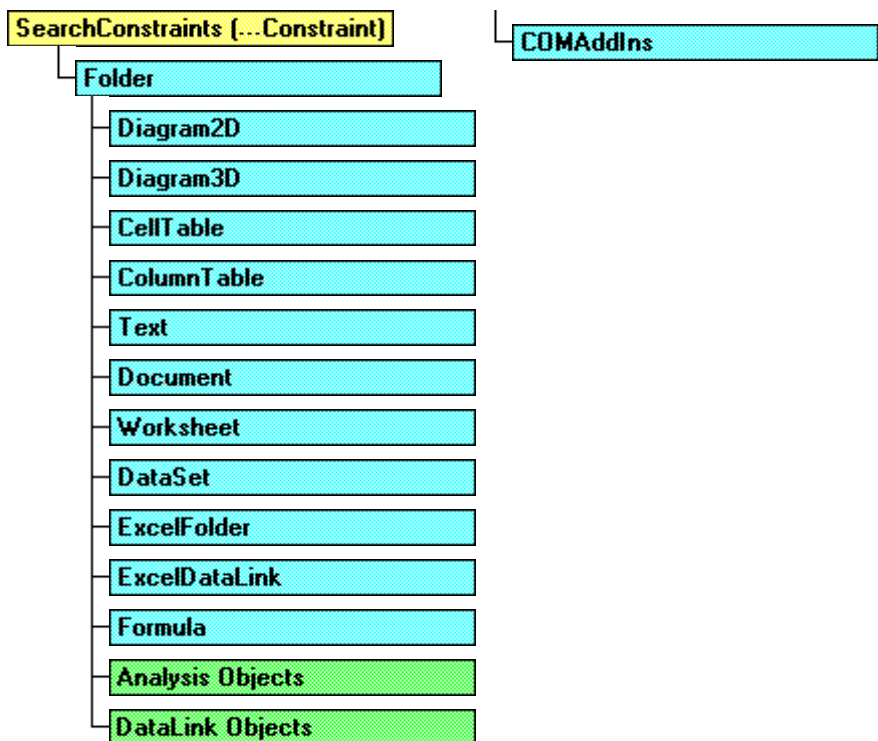
If in (1) **exp1** returns True, **exp2** will no longer be evaluated and the code in the If statement will be executed immediately. In (2) **exp2** is evaluated regardless of whether **exp1** returns TRUE or not. If **exp1** must return TRUE to validate **exp2**, the statement in VBA has to be disassembled:

```
If exp1 Then
    ...
    If exp2 Then
        ...
    End If
    ...
End If
```

### The FlexPro Object Model

#### Application



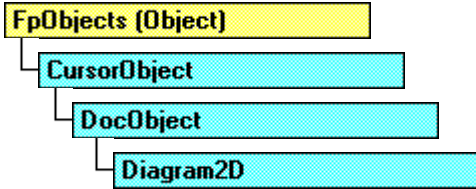


### Legend

- Object and Collection
- Object only
- Virtual object, only for structuring

### Understanding the FlexPro Object Hierarchy

As opposed to most Automation object models such as that of Microsoft Office, FlexPro objects are structured hierarchically. This means that properties and methods that apply to several objects are combined to form a base object. Structuring is multi-layered, ranging from general to specific. The object hierarchy of FlexPro applies a powerful principle of object-oriented programming, called inheritance. The following illustration shows the object hierarchy of the FlexPro 2D diagram.

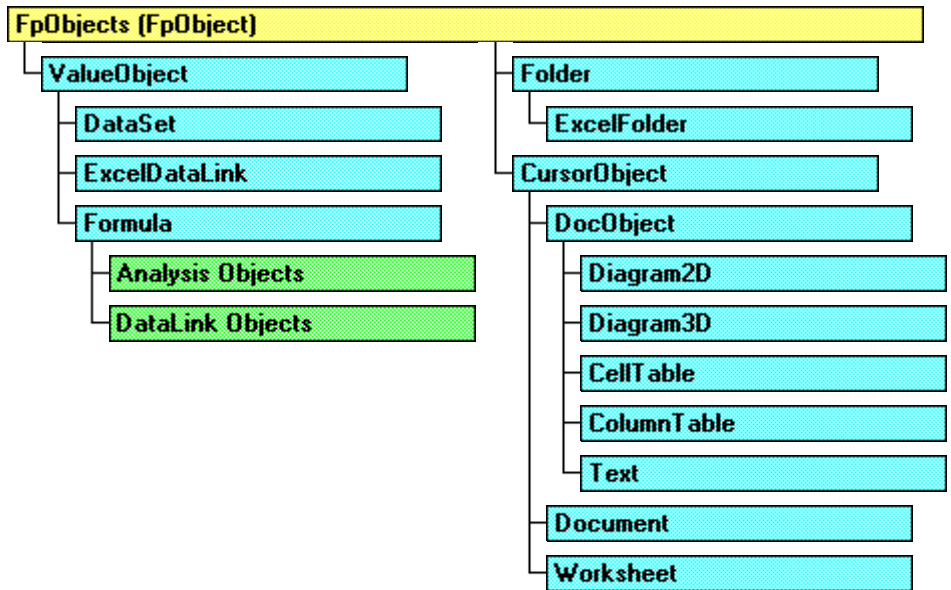


Inheritance offers a number of advantages with regard to both implementation of FlexPro and use of the FlexPro object model. For the implementation of FlexPro, inheritance means that the implementation of a base object can be re-used in all objects that are based on this base object. The software is therefore more compact and easier to maintain. In the above, example, the **CursorObject** is the base object for the 2D diagram (**Diagram2D**, the 3D diagram (**Diagram3D**), the worksheet (**Worksheet**) and the document (**Document**). One implementation is being used for four different objects. The further up you move in the object hierarchy, the greater this so-called "code re-use". For example, all objects that can be stored in the FlexPro database are derived from FpObject.

In addition, during the development of automation applications that access the FlexPro object model, the FlexPro object hierarchy represents a major advantage. For example, a program that only uses the properties and methods of the **CursorObject** interface will run with all FlexPro CursorObjects (2D diagram, 3D diagram, etc.) without requiring changes even if later versions of FlexPro contain additional objects derived from **CursorObject**. The compiler (Basic, C++, Java ...) used for the implementation of the application can generate very fast code because all methods and properties of **CursorObject** are known at compile time. Thus, calls can be directly converted to addresses, which considerably accelerates the subsequent processing of the program. In Basic you can benefit from this by choosing the **CursorObject** type for the object variable. You therefore tell the compiler which set of properties and methods will be available at the time the program is compiled.

If FlexPro did not have an object hierarchy, an object variable that is to have any object that supports cursors would have to be declared an unspecified **Object** type. Such a program could also work with new objects supporting cursors without requiring changes. The processing speed, however, would be much lower. Since the compiler does not know the properties and methods of the actual object held by the object variable during compiling of the program, the address, such as that of a method, cannot be established until the program is executed. Using the name of the method, its index in the interface is identified. The method is then called using the generic **Invoke** function. All arguments have to be packed in what are called variants, since their data types are not known at the time of compilation. Another serious disadvantage of this procedure is that many errors, such as "wrong method name," "wrong number of arguments" or "wrong argument data type," will not be identified until the program is executed.

## Overview of FlexPro's Object Hierarchy



### Legend

- Object and Collection
- Object only
- Virtual object, only for structuring

## Working with Objects

### Working with FlexPro Project Databases

FlexPro stores all objects in a project database. You can apply a hierarchical structure to this project database by creating folders. Thus, a FlexPro project database resembles a hard disk on your computer. Instead of files on the hard drive, a FlexPro project database contains data sets, formulas, presentation objects, worksheets and documents that you need for your analysis. Just like on a hard drive, you can build up a hierarchy of folders in a FlexPro project database to organize your FlexPro objects.

## Creating a Project Database

FlexPro can open multiple databases at once that are then stored in the **Databases** collection. To create a new, empty database, use the **Add** method of the **Databases** collection. The **Add** method provides the newly created database in such a way that you can assign it to a variable for later use:

```
Dim Doc As Database
Set Doc = Databases.Add
```

## Opening a Database

Use the **Open** method of the **Databases** collection to open a database stored in a file.

```
Dim Doc As Database
Set Doc = Databases.Open("C:\Databases\MyDatabase.fpd", fpOpenModeBackup)
```

## Saving a Database

If the database is new and has not been saved yet, use the **SaveAs** method. The second argument determines how the database is to be saved.

```
Doc.SaveAs "C:\Databases\MyDatabase.fpd", fpSaveModeOneFile
```

Use the **Save** method to save a database that has already been opened as a file and has been saved at least once before.

```
Doc.Save
```

Use the **Save** method of the **Databases** collection to save all databases that are currently open. For those databases that do not have a file name yet, a dialog box appears, in which you can specify the name.

```
Databases.Save
```

## Closing a Database

You can close an individual database using its **Close** method.

```
Doc.Close fpSaveChanges
```

Alternatively, you can close all databases that are currently open.

```
Databases.Close fpPromptToSaveChanges
```

## Accessing Databases

To access a database in the **Databases** collection, use the path name or the name that



is displayed in FlexPro's main window.

```
Set Doc = Databases("C:\Databases\MyDatabase.fpd")
```

Use the **ActiveDatabase** method of the **Application** object to access the currently active database.

```
ActiveDatabase.Close
```

## Creating Objects and Folders

Use the **Add** method of the **Folder** object to create objects and folders in the database.

Use the **RootFolder** property of the **Database** object to access the root folder.

```
Dim Fld As Folder
```

```
Set Fld = ActiveDatabase.RootFolder.Add("Folder", fpObjectTypeFolder)
```

The Add method can also add other objects, returning the new object as the result.

```
With Fld.Add("Series", fpObjectTypeFormula)
```

```
    .Formula = "(100, 0.0, 0.1)"
```

```
End With
```

## Accessing Objects

To access an object in a folder, use the **Object** method of the folder. You can specify the object name and object type or just the name with the extension related to the specific type.

```
ActiveDatabase.RootFolder.Object("Diagram", fpObjectTypeDiagram2D).Comments = "Measurement 1"
```

or

```
ActiveDatabase.RootFolder.Object("Diagram.2D").Comments = "Measurement 1"
```

Using the **Parent** property, which is available to every FlexPro object, you can access the object (usually a folder) that contains the object. The following example accesses the comment added to a folder containing the data set that has been assigned to the "Dataset" object variable.

```
Dataset.Parent.Comments = "Measurement 1"
```

## Searching the Database for Objects

Use the **Objects** method of the **Folder** object to search for objects. Several options are available.

The following example finds all data set objects that have names beginning with "sig".

```
Dim Objects
```

```
Set Objects = ActiveDatabase.RootFolder.Objects("^sig*.dat$")
```

The regular expression is structured as follows:

Sequence	Meaning
"^sig"	Specifies that the object name must start with "sig".
".*"	Represents a string of any length, consisting of any characters.
".dat\$"	Specifies that the object name must end in ".dat", i.e. that the object must be a data set.

The following example finds a dataset object called "Signal" and a 2D diagram called "Plot".

```
Dim Objects
Set Objects = ActiveDatabase.RootFolder.Objects("Signal", "Plot.2d")
```

The following example finds all objects in which the value of the **Origin** property is "Measurement 1".

```
Dim Constraints As New SearchConstraints
With Constraints.Add
    .SearchItem = fpSearchItemOrigin
    .CompareOperation = fpSearchCompareOpTextIs
    .SearchValue = "Measurement 1"
End With
Dim Objects
Set Objects = ActiveDatabase.RootFolder.Objects(Constraints)
```

Use the **Objects** method of the **Database** object to access objects in different folders simultaneously. The following example deletes two objects.

```
ActiveDatabase.Objects(Array("\Diagram.2d", "\Measurement1\Signal")).Delete
```

Use the **Search** method of the **Folder** object to search through all or parts of the database for objects. The following example searches for all data sets of which the maximum value is larger than 10 and moves these data sets to the "Data Folder" folder.

```
Dim Constraints As New SearchConstraints
With Constraints.Add
    .SearchItem = fpSearchItemObjectType
    .CompareOperation = fpSearchCompareOpObjectTypeIsKindOf
    .SearchValue = fpObjectTypeDataSet
End With
```

```
With Constraints.Add
```

```
    .SearchItem = fpSearchItemMaximum
```

```
    .CompareOperation = fpSearchCompareOpValueGreater
```

```
    .SearchValue = 10#
```

```
End With
```

```
ActiveDatabase.RootFolder.Search(True, Constraints).MoveTo ActiveDatabase.Object("\Data\
```

## Working with Data Sets

The FlexPro data sets are largely determined by these four properties: **DataStructure**, **DataType**, **NumberOfRows** and **NumberOfColumns**. The **DataStructure** property determines the data structure of the data set. FlexPro offers support for nine data structures, six of which are composed of up to three components: The **DataType** property determines the data type of the data set or, if the data set has an aggregate data structure, the data types of the individual components it contains. FlexPro also supports several data types for numbers, text, Boolean values, times and duration. Not all data types are allowed for all components of a data set. Strings, for example, cannot be used as a data type for data sets that have an aggregate data structure. The two properties **NumberOfRows** and **NumberOfColumns** determine the size of a data set. **NumberOfRows** can be used with all data structures except scalar values and determines the number of data rows. **NumberOfColumns** is only relevant for the two-dimensional data structures data matrix and signal series, and determines the number of columns. For all other data structures, this property has the value One and cannot be changed. A change in the **NumberOfRows** or **NumberOfColumns** may effect several components of a data set. If, for example, you increase the number of rows in a signal, its X and Y components will increase.

## Creating a Data Set

The best practice is to use FlexPro to generate the right code for the creation of a data set by activating the macro recorder and then creating a data set using the wizard. For example, for a signal series with three columns containing 100 values each, the code should look something like this:

```
With ActiveDatabase.RootFolder.Add("SignalSeries", fpObjectTypeDataSet)
```

```
    .DataStructure = fpDataStructureSignalSeries
```

```
    .DataType(fpDataComponentY) = fpDataTypeFloat64
```

```
    .DataType(fpDataComponentX) = fpDataTypeFloat64
```

```
    .NumberOfColumns = 3
```

```
    .NumberOfRows = 100
```

```
.FillColumns "(NumberOfRows(i), FloatingPoint64 0, FloatingPoint64 0)", _  
    fpDataComponentY  
.FillColumns "(NumberOfRows(i), FloatingPoint64 0, FloatingPoint64 1)", _  
    fpDataComponentX  
.Update
```

End With

An important aspect in creating the data set efficiently is to apply the properties **DataStructure**, **DataType**, **NumberOfRows** and **NumberOfColumns** in the correct sequence. Since the data set is restructured with every instance of write access to one of these properties, this data set should only reach its full size when it is last accessed. The wizard uses the correct sequence. First, the data structure of the data set is determined. Now the components of the data set are available, and their data types can be determined. Finally, the number of columns is specified. The data set only consists of one row containing three columns, i.e. three values. Once the number of rows is specified, the data set grows to include 300 values. Finally, the contents of the columns are initialized. This step can be omitted if the data set is to be filled with data at a later time.

### Reading and Writing the Data of a Data Set

If you use the **Value** method of the **DataSet** object without specifying arguments you can access the complete content of the data set. This approach is recommended for smaller data sets only, since the complete contents have to be transferred into a variant array, i.e. the main memory. If the data set has an aggregate data structure, **Value** passes all components as the signal object. If you want to assign a variable in Visual Basic, you have to use the **Set** statement. The following example reads a signal and then calculates the Delta-t from the X component.

```
Dim S As Signal  
Set S = SignalDataSet.Value  
Dim Delta As Double  
Delta = S.X(1) - S.X(0)
```

Please note that the first X value in S.X has the index zero. During write-access to the Value property, you also have to consider that the value with the index zero is contained in the array.

```
Dim V(2) As Double  
V(0) = 1  
V(1) = 2  
V(2) = 3
```

```
DataSet.Value = V
```

Thus, an array that is dimensioned with (n) in Basic has n + 1 values!

FlexPro can effectively process data sets of several gigabytes in size. This is possible because data sets that are a certain customizable size or larger are managed as files on the hard disk and therefore do not have to be loaded completely into main memory. When accessing the data of a data set via the Automation Object Model, the data to be transferred are always moved as a variant array and therefore copied completely into main memory. With very large data sets, you should therefore only access parts of the data. By using the optional arguments **Row** and **Column** of the **Value** property, you can directly access individual values, rows or columns in a data set.

The following example accesses the values in a rectangular area in the Y component of a signal series and sets these to zero.

```
Dim Row, Col As Long
For Row = 1 To 4
    For Col = 1 To 2
        SignalSeriesDataSet.Value(fpDataComponentY, Col, Row) = 0
    Next
Next
```

You can also transfer data block by block. To do this, you can use the **ValueObjectRange** object, which you can access using the **Range** property of the data set. This enables read/write access to a section of the data in the data set.

The following example copies the first five values in the Y component of a signal to the positions 6 to 10.

```
SignalDataSet.Value(fpDataComponentY, , 6, , 10).Value = SignalDataSet.Range(fpDataComp
```

#### Please note the following:

- Read and write access to the **ValueObjectRange** object are passed on directly to the data set. The **ValueObjectRange** object does not keep a copy of the data in the section.
- The **ValueObjectRange** object can also be used for formulas, data link objects and analysis objects. In this case, however, only read-access is allowed.
- By using the **Value** property of the **ValueObjectRange** object, not only can you access the complete range, but also columns, rows or individual values as well.
- The sequential indices of the **ValueObjectRange** object always start with One, even if the range in the data set starts with a higher index.

```
Dim Range As ValueObjectRange
```

```
Set Range = SignalSeriesDataSet.Range(fpDataComponentY, 1, 2, 2, 5)
Dim Data As Variant
Data = Range.Value 'Data is now of type variant/double(0 to 1, 0 to 3)
```

### Transferring Large Data Sets Section by Section

Saving data that is generated continually as a data set, e.g. during a measuring process, is very common. The obvious solution for this problem is to create the data set with the correct data structure and data type and to initialize the number of rows with zero. Upon receiving the data, the number of rows is then increased; a **ValueObjectRange** object is used to enter the most recent value into the data set. For a data series, the code for this procedure looks as follows:

```
With DataSeries
    .NumberOfRows = .NumberOfRows + 1
    .Value( , , .NumberOfRows) = NewValue
End With
```

The disadvantage of this procedure is, however, that the processing speed is comparatively low. With every increase of the data set, FlexPro has to copy the values internally, which is very expensive, particularly with larger data sets. Furthermore, the data volume transferred per each access to the Automation interface of FlexPro is very low. You can avoid these problems by first gathering the generated data in an array and then transferring the content of the array as a whole.

```
Dim Data(1000000 - 1) As Double
For i = 0 to 100000 - 1
    Data(i) = NewValue
Next
With DataSeries
    .NumberOfRows = .NumberOfRows + 1000000
    .Range(fpDataComponentY, , .NumberOfRows - 1000000 + 1, , .NumberOfRows).Value = Data
End With
```

## Communicating with Other Applications

Apart from working with FlexPro data, you might want your application to exchange data with other applications, such as Microsoft Excel. You can communicate with other applications by using Automation (formerly OLE Automation).

### Automating FlexPro From Within Another Application

With Automation, you can return, edit and export data by addressing objects, properties and methods of another application. Application objects that can be addressed by another application are called Automation objects.

To make FlexPro available to another Automation application, the first thing you have to do is create an Automation object that references the FlexPro Application object.

```
Set app = CreateObject("FlexPro.Application")
```

This statement makes the Application object in FlexPro available for Automation. You can control FlexPro by using the objects, properties and methods of the FlexPro Application object. The following statement, for example, creates a new FlexPro database.

```
app.Databases.Add
```

The function **CreateObject** starts a FlexPro session that is not closed by Automation if the object variable addressing the **Application** object becomes invalid. If the object reference is set to the keyword **Nothing**, FlexPro is not closed. Instead, use the **Quit** method to close the FlexPro application. In the following example, the FlexPro start path is shown. The **Quit** method is used to close the new instance of FlexPro after the start path has been displayed.

```
Set app = CreateObject("FlexPro.Application")
```

```
MsgBox app.Path
```

```
app.Quit
```

### Automating Another Application From Within FlexPro

If you want to use Automation from within FlexPro to exchange data with another application, the function **CreateObject** or **GetObject** provides a reference to the application. Then, use the objects, properties and methods of the other application to add, change or delete data. When you have finished making your changes, close the application. In the following FlexPro example, the Microsoft Excel start path is displayed. You can use the **Set** statement with the **Nothing** keyword to delete an object variable. This has the same effect as closing the application.

```
Set myobject = CreateObject("Excel.Application")
```

```
MsgBox myobject.Path
```

```
Set myobject = Nothing
```

For information on referencing an object library and using the objects it contains, go to Understanding Automation and Set a Reference to a Type Library.

### Programmatic OLE Identifiers

You can use a programmatic OLE identifier (sometimes also called ProgID) to create an automation object. The following tables show the programmatic OLE identifiers for the ActiveX control elements and FlexPro.

#### ActiveX control elements

If you want to create the ActiveX control elements listed in the following table, use the corresponding programmatic OLE identifier.

Control Element	Identifier
CheckBox	Forms.CheckBox.1
ComboBox	Forms.ComboBox.1
CommandButton	Forms.CommandButton.1
Frame	Forms.Frame.1
Image	Forms.Image.1
Label	Forms.Label.1
ListBox	Forms.ListBox.1
MultiPage	Forms.MultiPage.1
OptionButton	Forms.OptionButton.1
ScrollBar	Forms.ScrollBar.1
SpinButton	Forms.SpinButton.1
TabStrip	Forms.TabStrip.1
TextBox	Forms.TextBox.1
ToggleButton	Forms.ToggleButton.1

#### FlexPro

To create the FlexPro objects listed in the following table, use the corresponding programmatic OLE identifier. If you use an identifier without the version number suffix, you will create an object of the latest FlexPro version available on the computer on which the macro is executed.



Object	Identifier
Application	FlexPro.Application, FlexPro.Application.8
Database	FlexPro.Database, FlexPro.Database.8
SearchConstraints	FlexPro.SearchConstraints, FlexPro.SearchConstraints.1

## Working with Events

### Using Events with the Application or Databases Object

To create an event routine for an event affecting the Application or Databases object, carry out the following three steps:

1. Declare an object variable in a class module so that the events will be answered.
2. Write the specific event procedures.
3. Initialize the declared object from another module..

### Declaring the Object Variable

Before you can write procedures for the **Application** or **Databases** object events, you have to create a new class module and declare an object of the **Application** or **Databases** type with events. Let us assume that a new class module called "EventClassModule" is created. The new class module contains the following code:

```
Public WithEvents App As FlexPro.Application
or
Public WithEvents Databases As FlexPro.Databases
```

### Writing Event Procedures

After the new object with events has been declared, it appears in the **Object** drop-down list box in the class module, and you can write event procedures for the new object. (If you select the new object in the "Object" list, the valid events for the object are listed in the **Procedure** drop-down list box.) Select an event from this list. An empty procedure is added to the class module.

```
Private Sub App_DatabaseModified(ByVal Database As Object)
End Sub
or
Private Sub Databases_BeforeDatabaseClose(ByVal Database As Object)
End Sub
```

## Initializing the Declared Object

Before the procedure can run, a connection has to be established between the declared object in the class module ("App" or "Databases" in this example) and the **Application** or **Databases** object. This connection can be set up using the following code from any module:

```
Dim X As New EventClassModule
Sub Register_Event_Handler()
    Set X.App = FlexPro.Application
    Set X.Databases = FlexPro.Databases
End Sub
```

Start the "Register\_Event\_Handler" procedure. Once the procedure has been executed, the App object in the class module references the FlexPro **Application** object and the database object with regard to the **Databases** object. The event procedures in the class module are then called as soon as the events occur.

## Using Events with the Database Object

The **Database** object includes support for the following events: DatabaseSaved, DatabaseCreated and DatabaseOpened. Procedures, which respond to these as well as the other available events, are written into the **ThisDatabase** class module (open the **Procedure** drop-down list box to view the complete list of events). You can create an event procedure by following these steps:

1. Double click on **ThisDatabase** in your template project or database project in the "Project Explorer" window. (In the folder view, **ThisDatabase** can be found in the **FlexPro Objects** folder.)
2. Select Database from the **Object** drop-down list box.
3. Select an event from the **Procedure** drop-down list box.

An empty sub-routine is added to the class module.

4. Add the Visual Basic instructions to be executed when the event occurs.

The following example shows a **DatabaseCreated** event procedure in a template project, which is executed when a new database is created.

```
Private Sub Database_DatabaseCreated(ByVal Database As Object)
    MsgBox "New database created"
End Sub
```

The following example shows a DatabaseModified event procedure in a database project

that runs only after the database has been changed.

```
Private Sub Database_DatabaseModified(ByVal Database As Object)
    MsgBox "Database modified"
End Sub
```

If an auto-macro exists in a database and template database, only that automatic macro, which is stored in the database, is executed. If an event procedure for a **Database** event exists in a database and template database, both event procedures are executed.

---

**Note:** For information on the creation of event procedures for the **Application** or **Databases** object, go to Using Events with the Application or Databases Object.

---

### Using Events with FlexPro Objects

To create an event routine for a FlexPro object event, perform the following three steps:

1. Declare an object variable in a class module so that the events will be answered.
2. Write the specific event procedures.
3. Initialize the declared object from another module.

### Declaring the Object Variable

Before you can write the procedures for the events of a FlexPro object, you need to create a new class module and declare an object of the desired type (e.g. FpObject) with events. Let us assume that a new class module called "EventClassModule" is created. The new class module contains the following code:

```
Public WithEvents oObject As FpObject
```

### Writing Event Procedures

After the new object with events has been declared, it appears in the **Object** drop-down list box in the class module, and you can write event procedures for the new object. (If you select the new object in the **Object** list, the valid events for the object are listed in the **Procedure** drop-down list box.) Select an event from this list. An empty procedure is added to the class module.

```
Private Sub ObjectToObject_ObjectModified(ByVal Object As Object)
End Sub
```

## Initializing the Declared Object

Before the running the procedure, a connection has to be established between the declared object in the class module "oObj" and the **FpObject** object. This connection can be set up using the following code from any module:

```
Dim X As New EventClassModule
Sub Register_Event_Handler()
    Set X.oObject = ActiveDatabase.RootFolder.Object("Formula.FML")
End Sub
```

Start the "Register\_Event\_Handler" procedure. Once the procedure has been executed, the oObj object in the class module references the desired **FpObject** object. The event procedures in the class module are then called as soon as the events occur.

## Using Events with Class Objects

To create an event routine for an event affecting all FlexPro objects of a certain type, the event procedure has to be entered into one of the object modules AnyCursorObject, AnyDocObject, AnyFolder, AnyFormula, AnyFpObject or AnyValueObject . You can create such an event procedure by following these steps:

1. For instance, double-click on **AnyFpObject** in your template project or database project in the "Project Explorer" window. (In the folder view, **AnyFpObject** can be found in the **FlexPro Objects** folder.)
2. Select **FpObjectClass** from the **Object** drop-down list box.
3. Select an event from the **Procedure** drop-down list box.

An empty sub-routine is added to the class module.

4. Add the Visual Basic instructions to be executed when the event occurs.

The following example shows a ObjectModified event procedure in a template project, which is executed if an **FpObject** object is changed.

```
Private Sub FpObjectClass_ObjectModified(ByVal Object As Object)
    MsgBox Object.Name & " was modified."
End Sub
```

If an event procedure for a **FpObject** event exists in a database and template database, both event procedures are executed.

**Note:** You can find out whether a certain FlexPro object belongs to one of the object classes **AnyCursorObject**, **AnyDocObject**, **AnyFolder**, **AnyFormula**, **AnyFpObject** or **AnyValueObject** in Overview of FlexPro's Object Hierarchy.

---

## Working with Macros

### Recording a Macro to Generate Code

If you are not sure which Visual Basic method or property to use, you can activate macro recording and execute the process manually. Macro recording converts your actions to Visual Basic code. Once your actions have been recorded, you can adapt the code to your needs. If, for example, you are unsure which property or method is used to activate the cursors in the current diagram, do the following:

1. From the **Tools** menu, select **Macro** and then click **Record**.
2. If necessary, change the default name for the macro and click **OK** to start recording.
3. From the **Cursors** menu, select **Activate Cursors**.
4. On the **Record** toolbar, click **Exit**.
5. From the **Tools** menu, select **Macro** and then click **Macros**.
6. Select the macro name (see step 2) and click **Edit**.

Take a look at the Visual Basic code that sets the property corresponding to the activation state of the cursors. Position the cursor on **CursorActive** and press F1, or click **Help**. Within a topic, you can then view examples and objects that support the **CursorActive** property (click on **Applies to**).

**Remarks** Recorded macros often use the **Selection** object to manipulate the elements selected. For example, the following statement carries out a recording when you assign the color red to the selected curve of a diagram.

```
ActiveDatabase.ActiveObject.Selection.LineFormat.Color = fpColorRed
```

If you do not want the recorded code to use the **Selection** object, you have to edit the object using its Properties dialog box while recording a macro. In this case, the following code is recorded:

```
ActiveDatabase.ActiveObject.Curves(1).Line.LineFormat.Color = fpColorRed
```

You can also edit the recorded macro manually. You can find information on how to do this under **Revising Recorded Visual Basic Macros**.

## Revising Recorded Visual Basic Macros

Macro recording is very useful when trying to find the Visual Basic methods and properties that you would like to use. If you do not know which properties and methods to use, activate macro recording and execute the procedure manually. Macro recording compiles your actions into Visual Basic code. There are, however, some restrictions to recording macros. The following cannot be recorded:

- Conditional branches
- Variable assignments
- Loop structures
- Custom forms
- Error handling
- Customizing the FlexPro user interface

To enhance your macros, you might want to edit the code that was recorded in your module.

## Removing the Selection Property

Macros created through macro recording often depend on the selection you make. At the beginning of most recorded macros, you will see "Selection". Recorded macros use the **Selection** property to return the **Selection** object. In the following example, a line is added to a data set by entering the value 0 into the gray field in the editor beneath the last line of the data set.

```
Sub Macro1()  
    ActiveDatabase.ActiveObject.Selection.ActiveValue = 0#  
    ActiveDatabase.ActiveObject.Selection.Move fpDataSelectionDown  
End Sub
```

Although this macro executes the task, it has the disadvantage that it only runs correctly if the data set is currently being displayed in the data editor and the entry field beneath the last line is selected. These two problems can be solved by editing the macro in such a way that it does not use the **Selection** object. Here is the revised macro:

```
Sub MyMacro()  
    ActiveDatabase.ActiveObject.NumberOfRows = ActiveDatabase.ActiveObject.NumberOfRows + 1  
    ActiveDatabase.ActiveObject.Value(fpDataComponentAll, 1, ActiveDatabase.ActiveObject.NumberOfRows)  
End Sub
```

The first statement adds a line to the active data set. The second statement uses the **Range** method to return a **Range** object that precisely represents the newly inserted

value. By assigning the value 0 to the **Value** property of this **Range** object, the newly inserted value is initialized. For more information on how to use the **Range** object, go to Working with Data Sets.

### Using With...End With

Macro statements that refer to the same object can be simplified by using the structure **With...End With**. For example, the following macro was recorded when the selected text was changed to red, a bold font and a 10-point font size.

```
Sub Macro1()
    ActiveDatabase.ActiveObject.Selection.Font.Bold = True
    ActiveDatabase.ActiveObject.Selection.Font.Size = 10
    ActiveDatabase.ActiveObject.Selection.LineFormat.Color = fpColorRed
End Sub
```

The **Selection** property is used with every statement to return a **Selection** object. The macro can be simplified in such a way that the **Selection** property is used only once.

```
Sub MyMacro()
    With ActiveDatabase.ActiveObject.Selection
        .Font.Bold = True
        .Font.Size = 10
        .LineFormat.Color = fpColorRed
    End With
End Sub
```

### Auto Macros

By giving the macro a specific name, you can run it automatically when you carry out certain operations (e.g. launching FlexPro or opening a database). FlexPro recognizes the following names as auto macros:

Macro Name	Execution
AutoExec	When launching FlexPro or loading a template database
AutoNew	When creating a new database
AutoOpen	When opening an existing database
AutoClose	When closing a database
AutoExit	When exiting FlexPro or closing a template database

Auto macros are recognized in code modules if these contain a procedure that is named after the auto macro. Just like all other macros, auto macros can be saved in the personal template database or in a project database.

In case of a naming conflict (i.e. several auto macros have the same name), an auto macro in the active project database takes precedence over a macro of the same name in the personal template database.

**Remarks** If you hold down the SHIFT key, auto macros are not executed. If, for example, you are creating a new database and the personal template database contains an AutoNew macro, you can prevent this AutoNew macro from being executed by clicking on **New Project Database** on the **File** menu while holding down the SHIFT key until the new database is displayed.

### Storing Values When a Macro is Finished

When a macro ends, the values contained in its variables are not automatically saved to a data carrier. If a value is to be preserved, it has to be saved outside of the macro before the macro finishes running.

Each FlexPro object in a FlexPro database, such as a data set, diagram or folder, has a **Parameters** list that can take any number of parameters. When working with FlexPro, you can edit this list of parameters by right-clicking with your mouse on an object, selecting **Properties** and then under the **Properties** dialog box, switching to the **Parameters** tab.

You can also use a macro to add parameters to an object, assign a different value to an existing parameter or retrieve the current value. In the following example, a parameter is added to the root folder of the active database.

```
ActiveDatabase.RootFolder.Parameters.Add "Charge", 54700
```

In the following example, the existing parameter is retrieved.

```
charge = ActiveDatabase.RootFolder.Parameters("Charge")
```

The following example assigns a different value to an existing parameter.

```
ActiveDatabase.RootFolder.Parameters("Charge") = 58000
```



## Working with Command Bars

### Overview of Command Bars

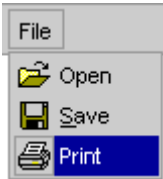
In FlexPro, toolbars and menu bars are controlled by the program as a common object type: as command bars. All of the following elements are represented in Visual Basic for Applications by **CommandBar** objects:

- Menu bars and toolbars
- Menus in menu bars and toolbars

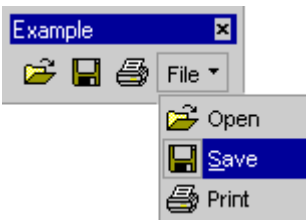
You can change all of the built-in menu bars and toolbars as well as create custom toolbars and menu bars for your Visual Basic application. The application functions are displayed as individual buttons on toolbars or as groups of commands in menus. The anchored toolbars in the following illustration contain three buttons in this example.



The menu in the following illustration contains the same three functions as in the illustration above, but this time they are menu commands.



In FlexPro, both menu bars and toolbars can contain menus. The floating toolbar in the following illustration contains three buttons and a menu in which the same functions are displayed as in the menu commands.



In Visual Basic, buttons are represented by **CommandBarButton** objects. Popup controls that display menus are represented by **CommandBarPopup** objects.

In FlexPro, you can also add text boxes, drop-down list boxes and combo boxes to any command bar. These three control types are represented in Visual Basic by **CommandBarComboBox** objects.

**Note:** Although they are similar in appearance and behavior, command bar controls and ActiveX controls are not the same. ActiveX controls cannot be added to command bars, and command bar controls cannot be added to documents or forms.

---

Built-in command bar controls in container applications are also represented in Visual Basic by **CommandBarButton**, **CommandBarPopup** and **CommandBarComboBox** objects, although their appearance and behavior can be more complex than that of the controls that you can add yourself. You can change the location and appearance of built-in controls, but not their default behavior. However, you have the option of assigning a custom macro to any built-in control and thus you can completely change its default behavior.

### Using Command Bars

In general, it is recommended that you use the functions for customization by FlexPro to create or change toolbars and menu bars that you want to have available using the Visual Basic application. Changes to toolbars and menu bars that are executed with FlexPro functions are described as design time changes.

You can add or change toolbars and menu bars (and their components) by using the CommandBars part of the FlexPro object model in Visual Basic code. You can create code that is executed once to create toolbars and menu bars. The code simulates changes that are performed during design time.

You can also create code that carries out changes to toolbars and menu bars in the Visual Basic application while FlexPro is running (e.g., for necessary disabling of a command in a command bar or for adding buttons to a toolbar as the reaction to particular user actions). Changes that are made within the context of the Visual Basic application are described as runtime changes.

### Adding and Managing Menu Bars and Commands

#### Adding Menu Items to the Menu Bar

If you want to add an entry to a FlexPro menu at runtime, use the **Add** method for the **CommandBarControls** collection. In the following example, an item is added to a menu.

```
Sub NewMenu()  
    Dim menubar As CommandBarPopup  
    Set menubar = CommandBars("Menu Bar").Controls.add(fpControlTypePopup)
```

```

With menubar
    .Visible = True
    .Caption = "newMenu"
End With
End Sub

```

## Modifying Menu Items at Runtime

You can make changes to the menu bar and its controls at runtime. Modifications that you make to the menu bar affect their appearance and placement. The following table lists the most frequently used properties and methods for changing menu bars at runtime.

Property or Method	Description
Add	Adds a menu bar by using the Add method from the <b>CommandBarControls</b> collection for the CommandBar called "Menu Bar". "fpControlTypePopup" must be selected as the type.
Enabled	If this property is set to True, the user can display the particular menu bar using Visual Basic code. If this property is set to <b>False</b> , the user cannot open the menu bar.
Visible	Specifies whether the control is displayed or hidden. If the control is hidden, the name of the menu bar still appears in the list of available toolbars.

You can add or remove elements, and you can determine the action that is performed using selected values. Built-in and custom functions can be set as the action for a control.

The following table lists the most frequently used properties and methods for changing the state, action or content of controls.

Property or Method	Purpose
Add	Adds a menu item to the toolbar. A menu item is of the <b>fpControlTypePopup</b> type.
Style	Specifies whether (and if so, how) the icon and/or label of an item is displayed. You can select one of the following <b>fpButtonStyle</b> constants: <b>fpButtonStyleAutomatic</b> , <b>fpButtonStyleCaption</b> , <b>fpButtonStyleIcon</b> , <b>fpButtonStyleIconAndCaption</b> , <b>fpButtonIconAndCaptionBelow</b> , <b>fpButtonIconAndWrapCaption</b> or <b>fpButtonIconAndWrapCaptionBelow</b> .

Property or Method	Purpose
OnAction	Specifies which procedure is performed when the user changes the value of the particular control.
Visible	Specifies whether the control is displayed or hidden to the user.

In the following example, the temporary popup control called **Custom** is added to the end of the active menu bar. The popup control called **Import** is then added to the **Custom** popup toolbar.

```
Sub NewMenu()  
    Dim myMenuBar As CommandBar  
    Dim newMenu As CommandBarPopup  
    Dim ctrl1 As CommandBarControl  
    Set myMenuBar = CommandBars.ActiveMenuBar  
    Set newMenu = myMenuBar.Controls.add(Type:=fpControlTypePopup, Temporary:=True)  
    newMenu.Caption = "Custom"  
    Set ctrl1 = newMenu.Controls.add(Type:=fpControlTypePopup, Id:=1)  
    With ctrl1  
        .Caption = "Import"  
        .TooltipText = "Import"  
        .Style = fpButtonStyleIconAndCaption  
    End With  
End Sub
```

## Adding and Changing Toolbars

FlexPro has extensive options for adding and designing custom toolbars. You can add built-in buttons, macros as buttons and even popup controls to toolbars. With changes during design time, which are usually made using Visual Basic code, you can also add or modify combo box controls. In addition, working with toolbars in code is almost exclusively limited to making changes at runtime (changing the state, appearance and actions of buttons, etc.).

## Changing Toolbars at Runtime

There are a number of changes that you can make to a toolbar during runtime. These include changes to the state of a command bar button on the toolbar. Every button control has the active states of being depressed (**fpButtonStateDown**) or not depressed (**fpButtonStateUp**). To change the state of a button control, you can use

the corresponding constant for the **State** property, as described in the table further down in this section.

The button appearance and action can also be changed at runtime. Use the **CopyFace** and **PasteFace** methods to change the button display (not your action). These methods are especially useful when you copy the picture from a particular button to the clipboard or if you want to import it into another application to change its characteristics. Use the **PasteFace** method to paste a button picture from the clipboard to a particular button.

To have a button action execute a function created by you, refer to the custom procedure names on the **OnAction** property of the button.

The following table lists the most frequently used properties and methods for changing the state, action or appearance of controls.

Property or Method	Description
CopyFace, PasteFace	Copies or pastes a button picture. Use the CopyFace method to copy the picture of the particular button to the clipboard. Use the PasteFace method to paste the clipboard contents to a particular button. The PasteFace method will not work if the clipboard is empty. If the picture in the clipboard is larger than the button, it will not be resized to fit the button. It is generally easier to copy and paste a button picture during design time. You can, however, also make changes to a button picture at runtime. Using the FaceId property, you can also assign built-in pictures to buttons.
Id	Specifies the value that represents the built-in functionality of the button. A button that updates a FlexPro object has an Id value of 32781, for instance.
State	Specifies the appearance or state of the button. This can be one of the following fpButtonState constants: fpButtonStateDown, fpButtonStateMixed or fpButtonStateUp.
Style	Specifies whether the picture or label is to be displayed on the button. This can be one of the following FpButtonStyle constants: fpButtonStyleAutomatic, fpButtonStyleIcon, fpButtonStyleCaption, fpButtonStyleIconAndCaption and fpButtonStyleIconAndCaptionBelow.
OnAction	Specifies which procedure is to be performed when the user clicks on a button, opens a menu or changes the content of a combo box control.
Visible	Specifies whether the control is displayed or hidden.
Enabled	Enables or disables a command bar.

In the following example, a new command bar is created with one button. By using the **OnAction** property of the **CommandBarButton** object, you can use the **CopyFace** method to copy the picture of the integrated **Open** button to the clipboard. You can

then paste the picture to the particular button using the **PasteFace** method. The appearance of the picture on the command bar button will then be changed during runtime.

```
Sub testAddModifyToolbars1()  
    Dim myBar As CommandBar  
    Dim oldControl As CommandBarControl  
    Set myBar = CommandBars _  
        .add(Name:="ChangingButton", Position:=fpBarPositionTop, Temporary:=True)  
    myBar.Visible = True  
    Set oldControl = myBar.Controls.add(Type:=fpControlTypeButton, _  
        Id:=CommandBars("Standard").Controls("Save database").Id)  
    oldControl.OnAction = "ActiveDatabase!changeFaces"  
End Sub  
  
Sub changeFaces()  
    Dim newControl As CommandBarControl  
    Set newControl = CommandBars.FindControl(Type:=fpControlTypeButton, _  
        Id:=CommandBars("Standard").Controls("Open database").Id)  
    newControl.CopyFace  
    Set oldControl = CommandBars("ChangingButton").Controls(1)  
    oldControl.PasteFace  
End Sub
```

### Adding and Changing Combo Box Controls

Text boxes, drop-down list boxes and combo boxes are new, powerful controls that you can add to toolbars in a Visual Basic application. For designing combo box controls, use the properties and methods listed in the following table.

Property or Method	Description
Add	Adds a combo box control to a command bar by specifying one of the following <b>fpControlType</b> constants for the Type argument: <b>fpControlTypeEdit</b> , <b>fpControlTypeDropDown</b> or <b>fpControlTypeComboBox</b> .
AddItem	Adds an element to the drop-down list of a drop-down list box or combo box. You can specify the index number of the new element in the list provided. However, if this number is higher than the number of elements in the list, then the <b>AddItem</b> method will not work.

Property or Method	Description
OnAction	Specifies which procedure is performed when the user changes the content of the combo box control.

In the following example, a combo box is added to a custom toolbar. The macro called **changeReport** is assigned to the combo box.

```
Sub ExampleCombo()
    Dim myBar As CommandBar
    Dim myCombobox As CommandBarComboBox
    Set myBar = CommandBars.add(Name:="Custom", Position:=fpBarPositionTop, Temporary:=True)
    Set myBar = CommandBars("Custom")
    With myBar
        .Controls.add Type:=fpControlTypeComboBox, Id:=1
        .Visible = True
    End With
    Set myCombobox = CommandBars("Custom").Controls(myBar.Controls.Count)
    With myCombobox
        .AddItem "Import Data"
        .AddItem "Update Report"
        .DescriptionText = "Create Report for Data"
        .Width = 200
        .OnAction = "ActiveDatabase!changeReport"
    End With
End Sub
```

While the application is running, the procedure assigned to the **OnAction** property of the combo box control is always called when the user changes the control. In the procedure, you can use the **ActionControl** property from the **CommandBars** object to determine which control was changed and to return the changed value. The **ListIndex** property returns the item that was entered or selected in the combo box.

```
Sub changeReport()
    Dim ctrlAction As CommandBarControl
    Dim myCombo As CommandBarComboBox
    Set ctrlAction = CommandBars.ActionControl
```

```
If TypeOf ctrlAction Is CommandBarComboBox Then
    Set myCombo = ctrlAction
    Select Case myCombo.ListIndex
        Case 1:
            MsgBox "1. Item"
        Case 2:
            MsgBox "2. Item"
        Case Else
            MsgBox "Other Item"
    End Select
End If
End Sub
```

### Adding and Showing Context Menus

Context menus are movable command toolbars that the user can display by clicking with the right mouse button. They can contain the same type of controls as command toolbars, and the controls behave in the same way as the command toolbar controls. In FlexPro, however, you cannot create or modify any context menus via the application interface. For this reason, you have to create or modify the context menus during runtime.

### Adding Context Menus During Runtime

The only difference between context menus and other command toolbars is that when creating the context menu, you must specify the **Add** method **fpBarPopup** as the **Position** argument. In the following example, a new context menu is created, and two controls (with labels) are added to it. The new menu then appears using the **ShowPopup** method.

```
Set copyAndPasteMenu = CommandBars.Add( _
    Name:="Custom", Position:=fpBarPopup, _
    Temporary:=True)
Set copy = copyAndPasteMenu.Controls.Add
With copy
    .FaceId = CommandBars("Standard").Controls("Copy").Id
    .Caption = "Copy the selection"
End With
```



```
Set paste = copyAndPasteMenu.Controls.Add
With paste
    .FaceId = CommandBars("Standard").Controls("Paste").Id
    .Caption = "Paste from the Clipboard"
End With
copyAndPasteMenu.ShowPopup 200, 200
```

## Showing Context Menus

Use the **ShowPopup** method to display context menus, as shown in the previous example.

If the container application supports the assignment of event procedures to user actions, you can display the context menu as a response to clicking with the right mouse button. However, not all applications support event procedures. Check the container application's documentation to find out if it supports event procedures.

## Modifying Context Menus During Runtime

Changes to context menus have to be made during runtime. The changes are generally limited to the appearance and action of the controls in the menu. You can read more about adding and managing menu commands under Adding and Managing Menu Bars and Commands.

## Examples

### A Custom Import Filter

This example implements a complete import filter for a binary file format. The resulting import filter supports virtually all options available for realizing custom import filters via Automation. It was specifically designed to be able to serve as the framework for your own import filters.

To keep the example simple, it does not include error handling.

### The Format to be Imported

The format to be imported is a binary format that was generated for illustrative purposes but features a common pattern. A program that can produce files in the sample format, and that was used to create the **Demofile.tst** sample file, is located as C++ source code in the same subfolder as the sample **ImportFilter.FPD** database and sample file. The path to the project database under Windows Vista or Windows 7 is usually C:\Users\Public\Documents\Weisang\FlexPro\10

\\Examples\\VBA\\ImportFilter\\ImportFilter.fpd or C:\\Users\\Public\\Public Documents\\Weisang\\FlexPro\\10\\Examples\\VBA\\ImportFilter\\ImportFilter.fpd.

You can implement a custom FPScript function in FPScript as well. See Custom FPScript Functions Tutorial.

The binary files consist of three parts:

- A file information structure that includes the number of channels and the sampling rate. Here is how it is defined in Visual Basic:

```
Private Type FileHeader
    strID As String * 8
    nVersion As Integer
    strOrigin As String * 32
    nNumberOfChannels As Long
    nNumberOfSamples As Long
    fSamplingRate As Double
    nTrigger As Long
```

End Type

- A number of channel header structures corresponding to the number of channels that describe the included channels:

```
Private Type ChannelHeader
    strName As String * 8
    strDescription As String * 64
    strUnit As String * 8
```

End Type

- The third part contains the data, starting with the data of the first channel, followed by those of the second, etc.

### Import Filter

To realize an import filter, a class module has to be created in a database; the module implements the **IImportFilter** interface (in the example: class module **DemoImportFilter**). By entering

Implements IImportFilter

in the code window of the class module, the entry **IImportFilter** becomes available in the object window. If this entry is selected, the two procedures of the interface **IImportFilter** are listed in the Procedure window. Once the two interface procedures

have been selected in the **Procedure** window, the corresponding procedure roots are added to the Code window, including the correct arguments.

### ImportSpy Procedure

Once a user has started an import procedure by calling the Import dialog box and selecting a file, the **ImportSpy** procedure of all import filters registered with FlexPro is initiated in order to find out which import filters can import the file.

In the example, **IImportFilter\_ImportSpy** first checks which filter was specified. This step only takes place for the sake of thoroughness. In this case it is not imperative because only one format has been registered for the import filter.

...

```
If Filter = m_strFilter Then
```

...

The filter verifies whether the name extension of the file corresponds to the one used for registering the filter:

...

```
If Right(UCASE(PathName), 4) = ".TST" Then
```

...

The file is not opened and the file header is not read until the verification yields a positive result:

...

```
Open PathName For Binary Access Read As #1 Len = Len(TheFileHeader)
```

```
Get #1, , TheFileHeader
```

```
Close #1
```

```
TheFileHeader.strID = CutString(TheFileHeader.strID)
```

```
If TheFileHeader.strID = "DEMO" And TheFileHeader.nVersion = 1 Then
```

```
    IImportFilter_ImportSpy = True
```

```
End If
```

...

The example file format contains an identification string and a version number that help to identify the format. If the identification string is correct and the file format features a version that is supported, the function will return **True**.

If additional specific import filters that support this file format exist, a selection dialog box appears during import in which the user can select the import filter to be used. FlexPro then calls the **Import** procedure of the import filter in order to import the file.

## Import Procedure

The **Import** procedure is only called if the **ImportSpy** procedure of the import filter has been called successfully.

As with the **ImportSpy** procedure, the first step is a verification of the filter. Then, the file header of the specified file as well as the header information for all elements or channels are read:

...

```
Open PathName For Binary Access Read As #1
```

```
Get #1, , TheFileHeader
```

```
For i = 1 To TheFileHeader.nNumberOfChannels
```

```
    Get #1, , TheChannelHeader
```

```
    Set oImportItem = New ImportItem
```

```
    oImportItem.strName = CutString(TheChannelHeader.strName)
```

```
    oImportItem.strDescription = CutString(TheChannelHeader.strDescription)
```

```
    oImportItem.strUnit = CutString(TheChannelHeader.strUnit)
```

```
    oImportItem.nSamples = TheFileHeader.nNumberOfSamples
```

```
    oItemColl.Add oImportItem
```

```
Next i
```

```
nDataStartPos = Seek(1) - 1
```

```
Close #1
```

...

The elements found are stored in a **Collection** object together with all the data relevant for presentation in the dialog box or for import. If the file is automatically imported in its entirety, a dialog box appears where the user can choose the elements to be imported.

...

```
If (Flags And fpImportOptionAutomatic) = 0 Then
```

```
    Set oFrm = New ImportItemsSelectFrm
```

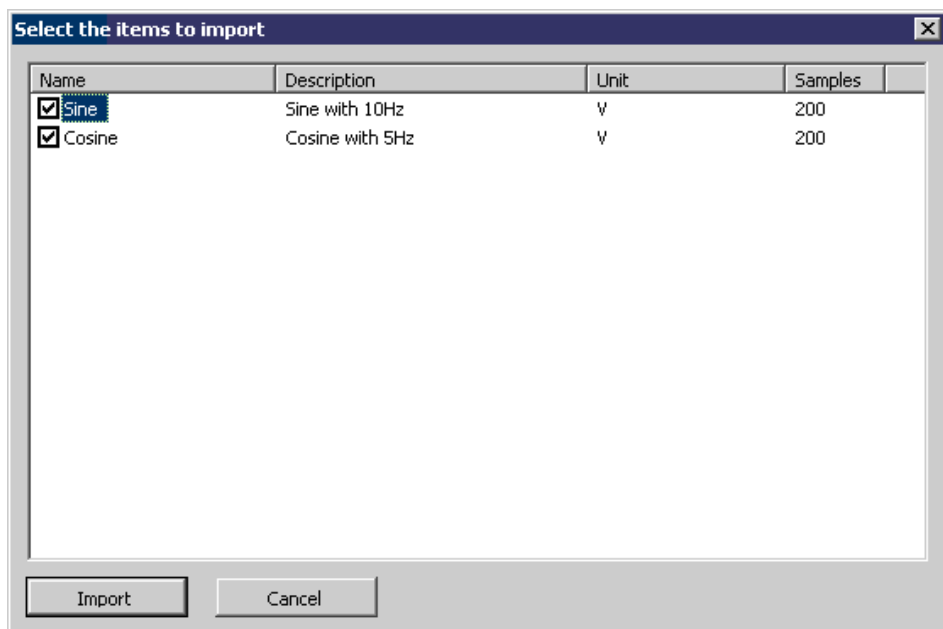
```
    oFrm.InitAndShow oItemColl, Flags
```

```
    bCancel = oFrm.m_bCancel
```

```
    Unload oFrm
```

```
End If
```

...



Eventually, the selected elements are then imported, taking into account the Flags parameter. If a new subfolder is to be created for each file to be imported, these subfolders are created first:

```
...
If Flags And fpImportOptionSubfolder Then
    Set oFile = oFS.GetFile(PathName)
    Set oImportFolder = Folder.Add(Left(oFile.Name, InStrRev(oFile.Name, ".") - 1), fp
Else
    ' import into the given folder
    Set oImportFolder = Folder
End If
```

...

If the data are not to be imported as signals, an X data set is created first; it can be referenced by all further data sets as an X component. In the example, there is only one sampling interval and one trigger, so that the X component can be calculated using a formula:

...

```
If (Flags And fpImportOptionSignal) = 0 Then
```

```
Set oXItem = oImportFolder.Add("XItem", fpObjectTypeFormula)
oXItem.Origin = TheFileHeader.strOrigin
oXItem.Component = fpDataComponentX
oXItem.Formula = "(" & CStr(TheFileHeader.nNumberOfSamples) & ", " & _
    CStr(-(TheFileHeader.nTrigger * oBinaryDataLink.SamplingInterval)) & ", " & _
    CStr(1 / TheFileHeader.fSamplingRate) & ")"

End If
...

Lastly, all selected Y components in the file are imported. In the example, this can be
done by taking advantage of the capabilities FlexPro binary import offers. For every
channel to be imported, a BinaryDataLink object is created and parameterized
according to the specifications. Lastly, the procedure decides whether the object created
is evaluated into a data set (the flag fpImportOptionLink was set) or whether the
BinaryDataLink object is preserved as a link.

...

For i = 1 To oItemColl.Count
    If oItemColl(i).bSelected = True Then
        Set oBinaryDataLink = oImportFolder.Add(oItemColl(i).strName, fpObjectTypeBinaryDataLink)
        ' general properties
        oBinaryDataLink.CommentsY = oItemColl(i).strDescription
        oBinaryDataLink.Origin = TheFileHeader.strOrigin
        oBinaryDataLink.Author = Application.UserName
        ' import properties
        oBinaryDataLink.FilePath = PathName
        oBinaryDataLink.NumberOfBlocks = 1
        oBinaryDataLink.BlockSize = 200
        oBinaryDataLink.ByteDistance = 0
        oBinaryDataLink.ResultDataType = fpBinaryDataLinkResultDataTypeFloatingPoint64
        oBinaryDataLink.DataType = fpBinaryDataLinkDataTypeFloatingPoint64
        oBinaryDataLink.ByteOffset = nDataStartPos + ((i - 1) * TheFileHeader.nNumberOfSamples *
        ' as signal ?
        If Flags And fpImportOptionSignal Then
            oBinaryDataLink.AsSignal = True
            oBinaryDataLink.SamplingInterval = 1 / TheFileHeader.fSamplingRate
```

```

        oBinaryDataLink.SamplingOrigin = -(TheFileHeader.nTrigger * oBinaryDataLink.SamplingRate)
Else
    ' assign the X component created above to this item
    oBinaryDataLink.Component = fpDataComponentY
    oBinaryDataLink.AssignedX = oXItem.Name
End If
' if import action is copy than evaluate the binary data link
If (Flags And fpImportOptionLink) = 0 Then
    oBinaryDataLink.Evaluate
End If
End If
Next i
...

```

If the import has been completed successfully, the procedure will return **True**.

### Registering an Import Filter

For a customized import filter to be available in FlexPro's **Import** dialog box (the file format is listed under **File Type**, it has to be generated using

```
Dim oImportFilter As New DemoImportFilter
```

and then be registered with the FlexPro **Application** object by calling **RegisterImport**.

```

RegisterImport oImportFilter.m_strFilter, _
    fpImportOptionSpecific Or _
    fpImportOptionLink Or fpImportOptionNoLink Or _
    fpImportOptionSubfolder Or fpImportOptionNoSubfolder Or _
    fpImportOptionAutomatic Or fpImportOptionManual Or _
    fpImportOptionSignal Or fpImportOptionNoSignal Or _
    fpImportOptionNoCalendarTime, _
    oImportFilter

```

During registration, the import filter itself is referenced and specifies which of the options available in the Import filter dialog box are supported.

If, for example, an import filter does not support links, only the **fpImportOptionNoLink** flag is specified in the first argument of **RegisterImport**. The **Create links** option will be grayed out and will not be set if this import filter is selected in the **Import** dialog box. If only **fpImportOptionLink** is specified, the option **Create links** is also gray, but activated. If both options are supported (

**fpImportOptionLink** or **fpImportOptionNoLink**), the user can freely choose how the import will take place.

This procedure also applies to the following options:

- Create new folder for each file
- Use calendar time
- Import as signals

If an import filter is to be available in all databases of one user or to all users in general, it is recommended that you implement it in the personal template database and to register it in the procedure **AutoExec**, which is executed automatically when the template database is loaded. De-registration can then be handled by the **AutoExit** procedure (see also **Auto Macros**).

---

**Note:** When developing a customized import filter, please consider the following: When a VBA project is reset, such as due to code changes in the debugger, previously registered import filters of this project will no longer be called. The import filter has to be de-registered and re-registered again.

---

## A Custom FPScript Function

This example implements a custom FPScript function that can be used in FlexPro FPScript formulas like a built-in function after registration. It is supported in the wizard of the FPScript Editor and is automatically available in all loaded databases.

---

### Notes:

The example shows how to implement this type of function in VBA. The complete VBA source text for the example is in the **CustomFunction.FPD** project database. However, you can also use any other automation-compatible programming language, such as C# or C++. A variant of the example that has been developed as an add-in in C++, is also included with FlexPro. The path to the project database is usually C:

```
\Users\Public\Documents\Weisang\FlexPro\10
\Examples\VBA\CustomFunction\CustomFunction.fpd or C:
>Users>Public>Public
Documents>Weisang>FlexPro>10>Examples>VBA>CustomFunction>CustomFunction
.fpd.
```

You can implement a custom FPScript function in FPScript as well. See **Custom FPScript Functions Tutorial**.

---



## Background

The example defines a function with three arguments that adds the first two arguments together and dependent on the third argument. It is constructed from several built-in functions. The example covers the essential elements for implementing a custom FScript function:

- Defining arguments
- Argument type and structure restrictions
- Using default arguments
- Registering and unregistering the function
- Accessing the arguments after the function call
- Defining custom FScript constants

## Registering a Custom Function

Before you can register a function, you have to define its arguments and other properties. The registration itself then appears at the end of the function definition.

First the function is added to the **UserDefinedFScriptFunctions** collection. You should then assign it a description. This is then automatically displayed when the wizard window is used, for instance.

```
With UserDefinedFScriptFunctions.Add("MyFunction")
    .Description = "Adds or subtracts two values"
    .Indeterministic = False
    ...
End With
```

The **Indeterministic** property defines whether the function always returns the same result for the same input. If **Indeterministic** is set to **True**, formulas that use this function have to be calculated at each update interval. Therefore, this value should be set to **True** only if absolutely required, such as when there are changing external dependencies.

After the function has been accepted into the **UserDefinedFScriptFunctions** collection, the required arguments can be added. Only the name of the argument is required, all other settings are optional. However, it is recommended that you provide a description so that a note regarding the argument can be displayed in the wizard window. In addition, the argument data types and structures are usually limited to the exact degree required. The advantage of this procedure is that FlexPro already handles checking for conformance with data type and structure restrictions before the custom

FPScript function routine is called. This makes it possible to limit the amount of code necessary for implementing the function and allows FlexPro to handle the output of error messages.

```
With .Parameters.Add("Arg1")
    .Description = "First argument"
    .AllowedTypes = fpParameterTypeNumeric
    .AllowedStructures = fpParameterStructureScalar Or _
        fpParameterStructureDataSeries Or fpParameterStructureSignal
End With
```

In the case above, the first argument is limited to numeric data types and scalar values, data series and signals are permitted as data structures. Data types and structures can be linked for the assignment with the OR operator as shown. If the argument deviates from the specified restrictions when the function is used, an error message will appear.

The default value is used for the last argument of the function. This means that the argument can be omitted. In this case, the default value for this argument can be used internally.

```
With .Parameters.Add("Operation")
    .Description = "Type of operation"
    .AllowedTypes = fpParameterTypeNumeric
    .AllowedStructures = fpParameterStructureScalar
    .DefaultValue = "MYFUNC_OPERATION_ADD"
End With
```

The case above has a unique feature. The type of **Operation** parameter is numeric, but text is specified as the default value. The text contains the name of a previously included custom FPScript constant whose value is determined automatically as a default value in this case. The advantage of this type of definition is that the more accessible name of the constant is displayed in the wizard window.

Custom FPScript constants can be inserted as follows:

```
With UserDefinedFPScriptConstants
    .Add "MYFUNC_OPERATION_ADD", "Selects the add operation", 1
    .Add "MYFUNC_OPERATION_MIN", "Selects the subtract operation", 2
End With
```

A new unit is simply added to the **UserDefinedFPScriptConstants** collection by specifying a name, description and scalar value. Conflicts with the names of existing built-in or custom constants will result in a relevant error message.

After the function parameters are declared, you can register the function for use. The **Register** method is called for this purpose. As an argument, it will receive a link to an object that implements the **IUserDefinedFunctionCalculate** interface.

```
.Register oMyFunction
```

The **IUserDefinedFunctionCalculate** interface includes the **Calculate** method that is called by FlexPro when calculating a formula that uses MyFunction, as long as it complies with the parameter restrictions.

---

**Notes:**

Only after registering a function are the defined parameters checked for consistency. If an error occurs during this check, you should review the code that defines the parameters for problems.

Typically, custom FScript functions are registered in an automatically executed function like **AutoOpen** or **AutoExec** so that they are available at a defined time early on.

---

Unregistering is the opposite of registering a function, so that usually in **AutoOpen** or **AutoClose**, the following occurs:

```
Sub AutoClose()  
    ' unregister function  
    UserDefinedFScriptFunctions.Item("MyFunction").Delete  
  
    ' unregister constants  
    With UserDefinedFScriptConstants  
        .Item("MYFUNC_OPERATION_ADD").Delete  
        .Item("MYFUNC_OPERATION_SUB").Delete  
    End With  
End Sub
```

Deregistration occurs automatically by deleting the function from the **UserDefinedFScriptFunctions** collection. Constants are unregistered in the same manner.

---

**Note:** Unregistering custom FScript elements is not absolutely necessary, since they are automatically removed once the program is closed. Since, however, just an object reference is supplied when a function is registered, explicitly unregistering it results in controlled termination of the object that executes the function calculation. This makes it possible to avoid unwanted side-effects (such as FlexPro hanging).

---

## Using a Custom Function

When calling the registered FPScript function **MyFunction** the **Calculate** method of the **IUserDefinedFunctionCalculate** interface is called. The arguments are passed in the form of a **Variance** array. The implementation example demonstrates how to access arguments. Due to the possible variety of data structures and types, the example has been simplified in some spots.

The array with the arguments **SafeArrayOfArguments** is one-based, as is typical in VBA. You can therefore access the desired operation by using index three:

```
' MYFUNC_OPERATION_ADD = 1
' MYFUNC_OPERATION_SUB = 2
```

```
nOperation = SafeArrayOfArguments(3)
```

The third argument has been restricted to scalar values of the numeric type. However, a void value can still be specified for the operation to be executed. Therefore, a check is carried out:

```
' check operation
If nOperation <> 1 And nOperation <> 2 Then
' will be propagated as error 0x800a0002 to FlexPro
    Err.Raise 2, "IUserDefinedFunctionCalculate_Calculate", "Invalid operation value"
    , "Invalid operation value"
End If
```

In the event of an error, an exception occurs. The message "Invalid operation value" then appears together with the specific error code either in the FlexPro Event Log or in an error dialog box.

The remaining part of the function calculation shows how you conduct case differentiations using the first argument with the help of the VB functions **IsObject**, **TypeOf** and **IsArray** and then execute the desired operation.

---

## Notes on using VBA for development

- If after registering a function, its **Calculate** method is changed, the function has to be unregistered and then reregistered. This is necessary because otherwise the entry point for the function is no longer valid and an error message will be displayed.
- At least when testing a **Calculate** method, you should make sure that under **Tools/Options** on the **System Settings** tab, the option **Update objects in the background** is disabled. Otherwise, FlexPro will crash if you use breakpoints in the **Calculate** method. The reason for this is that problems occur in the VBA environment

when calling the **Calculate** method in the background program thread.

- For custom FPScript functions in VBA, the runtime environment ensures that the **Calculate** method runs only once at a certain time. This limitation is in place for reasons of safety. For one thing, the VBA runtime environment does not support multiprocessing very well, and for another, the necessary limitation to one running instance in VBA is not possible using the language as the sole medium.

## Displaying a Diagram on a Form

This example illustrates a visualization of measured or calculated low-frequency data.

Every second, a new value pair (time plus measured value) is added to a signal. After that, a diagram, which contains this signal as a curve, is updated and also updated in the form. As long as the data set with the acquired values contains less than 100 values, the complete signal is displayed. Otherwise, only the last 100 values are shown.

The example code and the corresponding FlexPro objects are stored in the **Visualize**.

**FPD** database. The path to the Project Database is usually C :

```
\Users\Public\Documents\Weisang\FlexPro\10
\Examples\VBA\Visualize\Visualize.fpd or C:>Users>Public>Public
Documents>Weisang>FlexPro>10>Examples>VBA>CustomFunction>CustomFunction
.fpd.
```

## Creating a Form for Diagram Display

The required form only contains an image control that handles the display of the diagram as well as a button for closing the form.

While the form is initialized, the API function **SetTimer** is used to initialize an interval timer, which calls the **UpdateProc** procedure once per second. During the termination phase of the form, the timer created is shut down using the **KillTimer** API function.

```
Private Sub UserForm_Initialize()
    '    init random number generator (needed for the sample data)
    Randomize
    nTimerID = SetTimer(0, 0, 1000, AddressOf UpdateProc)
End Sub

Private Sub UserForm_Terminate()
    KillTimer 0, nTimerID
End Sub
```

## Inserting and Updating the Diagram

In the **UpdateProc** procedure,

On Error Resume Next

is used to prevent the procedure from being aborted if an error occurs. Use of the **SetTimer** API requires this to make sure that the procedure call always returns to Windows. After that, the procedure verifies whether the target signal for the data is already available in the root folder of the database. If that is not the case, it is created and initialized; otherwise, a new row is added to the signal.

If oSignal Is Nothing Then

```
Set oSignal = ActiveDatabase.RootFolder.Add("Signal", fpObjectTypeDataSet)
oSignal.DataStructure = fpDataStructureSignal
oSignal.DataType(fpDataComponentX)= fpDataTypeCalendarTime
oSignal.DataType(fpDataComponentY) = fpDataTypeFloat64
' we now have a signal with one value
```

Else ' increase the rows of the signal

```
oSignal.NumberOfRows = oSignal.NumberOfRows + 1
```

End If

The new value is then calculated and appended to the existing signal using the **Range** object (see also Working with Data Sets).

' you could acquire the data here, in this sample we calculate

' the new value pair

```
oSignal.Value(fpDataComponentX, , oSignal.NumberOfRows) = Now
oSignal.Value(fpDataComponentY, , oSignal.NumberOfRows) = (fMax - fMin + 1) * Rnd + fMin
```

At the end of the function, the diagram to be shown in the form is updated and the **Picture** property of the diagram is assigned to the **Picture** property of the viewing control field in the form.

```
Set oDiag = ActiveDatabase.RootFolder.Object("2D-Diagram.2D")
```

```
oDiag.Update
```

```
Display.DiagramImage.Picture = oDiag.Picture
```

## FlexPro Objects Required for the Visualization

To be able to visualize the data as described above, a diagram and a formula object were created in the root folder of the database. The diagram serves as the basis for the visual presentation in the form. It contains exactly one curve consisting of a signal that

provides the **LastValuesOfSignal** formula. The formula verifies whether the number of values in the signal is larger than the customizable maximum (100 in the example). If this is the case, the formula returns the most recent 100 values of the signal; otherwise, it returns the complete signal.

```
nMax = 100
nCount = NumberOfRows(Signal)
if nCount < nMax then
    return Signal
else
    return Signal[-nMax, -1]
end
```

## Processing Events

This example shows how you can capture and respond to events in your Basic programs. The program code and the required FlexPro objects can be found in the database called `Events.fpd`. This database contains a diagram with two X axes. The X values of the curve are scaled above the lower axis. In the example, the curve represents a route measured in km. The second axis, which is shown at the top of the diagram, assigns names to certain locations along this route. In the example, these are names of cities that can be found on the route. Two data series were created to scale this axis. The **Cities** data series contains the names of the cities, and the **Positions** data series contains their locations along the route. To label the axis with this data, the **Linear, division by data set** axis type was selected on the **Axis Scaling** tab and the **Positions** data series was entered as a data set. This determines the points at which axis divisions are to appear. On the **Division Labeling** tab, the option **Division labeling using a data series, signal or function** was enabled and **Cities** was entered as a data set. Since FlexPro 7 you have been able to easily synchronize the axis end values of this second X axis with the first on the **Axis Scaling** by selecting the option **Same as previous axis**. This axis cannot be auto-scaled because no curve will be shown above it.

The following example, which was originally developed for FlexPro 6, illustrates how you can synchronize the end values of the second axis with the first by processing events based on the following requirements:

- Event processing must be activated when you open the database and reset to its former state when you close the database.
- When opening the diagram, the second axis is to be set to fixed scaling and the current end values of the first axis are to be adopted.

- When zooming in or out using the cursor, the new end values of the first X axis are to be adopted by the second.

### Activating Event Processing

This task can be executed by implementing the two auto macros **AutoOpen** and **AutoClose**. The **AutoMacros** module contains the following code:

```
Private bEnableEvents As Boolean
```

```
Sub AutoOpen()  
    bEnableEvents = Application.EnableEvents  
    Application.EnableEvents = True  
End Sub
```

```
Sub AutoClose()  
    Application.EnableEvents = bEnableEvents  
End Sub
```

**AutoOpen** stores the current status of the application object's **EnableEvents** property and then event handling is enabled.

The previous state is restored in **AutoClose**.

### Initializing Axis Scaling

This task is executed by an event procedure that handles the **BeforeOpenObject** event, which is sent by every FlexPro object before opening it. In the **VbaProject** of **Events.fpd** there is a corresponding object called **AnyFpObject** which represents all FlexPro objects of the database and therefore receives all events common to all FlexPro objects. The following code is assigned to this class object:

```
Private Sub FpObjectClass_BeforeOpenObject(ByVal Object As Object)  
    Dim oDiag As Diagram2D  
    If Object.FullName = "\Route.2D" Then  
        Set oDiag = Object  
        oDiag.XAxes(2).Scaling.ModeStartValue = fpAxisScalingModeFixed  
        oDiag.XAxes(2).Scaling.ModeEndValue = fpAxisScalingModeFixed  
        oDiag.XAxes(2).Scaling.StartValue = oDiag.XAxes(1).Scaling.CurrentStartValue
```



```

        oDiag.XAxes(2).Scaling.EndValue = oDiag.XAxes(1).Scaling.CurrentEndValue
    End If
End Sub

```

The `If` test verifies whether the correct 2D diagram has been selected. The name, database path and the file name extension for the 2D diagrams have to match. The event procedure receives this event when any database object, e.g. another 2D diagram or data set, is opened. Within the `If` test, it is clear that we are dealing with a 2D diagram. The undefined object can therefore be assigned to a variable of the **Diagram2D** type. The next steps are easy to understand. At first, the starting value and the end value are set to fixed scaling. After that, the current starting and end values of the first axis are adopted. The properties **CurrentStartValue** and **CurrentEndValue** provide the values resulting from auto-scaling and not those that have been entered on the **Axis Scaling** tab of the axis, which are inactive in this case.

### Adopting the Axis End Values After Zoom Operations

Here, a multitude of events have to be handled, since several operations affect the axis end values. You can zoom in between the cursors, zoom in on a rectangular area or scroll along the axis. All these events only apply to cursor objects, i.e. objects that may contain cursors, which are diagrams, worksheets and documents. The **AnyCursorObject** class object, which receives all cursor events, represents all cursor objects. It contains the following code:

```

Private Sub AdaptScaling(Object As CursorObject)
    Dim oDiag As Diagram2D
    If Object.FullName = "\Route.2D" Then
        Set oDiag = Object
        oDiag.XAxes(2).Scaling.StartValue = oDiag.XAxes(1).Scaling.CurrentStartValue
        oDiag.XAxes(2).Scaling.EndValue = oDiag.XAxes(1).Scaling.CurrentEndValue
    End If
End Sub

Private Sub CursorObjectClass_AxisScaled(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub

```

```
Private Sub CursorObjectClass_AxisScrolled(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub

Private Sub CursorObjectClass_RectangleZoomed(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub

Private Sub CursorObjectClass_ZoomedBetweenCursors(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub

Private Sub CursorObjectClass_ZoomReset(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub

Private Sub CursorObjectClass_ZoomUndone(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub
```

**AdaptScaling** is an auxiliary procedure called by the different event procedures. Again, the procedure first verifies that the object to be processed does exist. The new axis end values are then copied. All event procedures with events that have an impact on axis scaling were implemented. And all events that are sent after the corresponding operation were selected so that the axis end values of the first X axis already feature the new values.

## 8.3 Additional Options

### Remote Control of FlexPro via FPAccess

#### FPAccess Interface

Apart from the Automation Object Model, FlexPro offers the FPAccess interface, which allows controlling programs such as an application for data acquisition or a Visual Basic program to access the project database externally and to remote control FlexPro in a limited way. The FPAccess interface has a much simpler structure than the Automation

Object Model and is targeted for efficient transfer of large volumes of data. The advantage of the FPAccess interface is that it is also available as a DLL and thus part of its functionality is available without FlexPro being installed.

The functionality of the FPAccess interface is a superset of that of FlexPro's DDE interface is therefore preferred. You can use the FPAccess interface to:

1. open FlexPro project databases,
2. manage FlexPro project databases in an Explorer dialog box,
3. create folders in FlexPro project databases,
4. write formulas and data sets in project databases,
5. check whether an object is present in the project database, and
6. open, update and print objects in project databases (only in direct communication with FlexPro - see below).

The interface can open FlexPro project database files independently without FlexPro's help, and can write data to these files. If the project database file has already been opened with FlexPro, then a direct communication with FlexPro will occur. DCOM allows this communication to take place across networks as well.

The project database access objects provided by FlexPro support OLE automation and direct access using COM. You can therefore also make best use of the interface with compiler languages such as C++.

Reference

### **Project Database Access for Acquisition Programs**

The FlexPro package contains modules with which you can prepare your acquisition software for direct access to FlexPro project database files. The following programs are currently supported:

- DasyLab Version 4.x - 10.x
- LabVIEW 6.x - 8.x
- Agilent VEE 5.0 - 8.x

The modules were tested using the most current acquisition software version number provided above. If you are using a newer software version, most likely the module will work with that version.

All components are based on the FlexPro FPAccess interface. The project database files can therefore also be opened independently, without running FlexPro. If the project database file has already been opened with FlexPro, then a direct communication with FlexPro will occur. This communication can also take place across networks. You can find

more details in the documentation covering the FPAccess Interface.

### Installation

The matching component may have already been supplied with your acquisition program. If this is not the case, then you will have to restart the FlexPro Setup program from the FlexPro CD.

1. For the Setup Type, select **Custom**.
2. If you purchased a single user license from FlexPro, you are only allowed to install FlexPro on one computer. Therefore, select **only** the **FPAccess Interface** option and the **Extension Modules** option further down.
3. Click **Next**. A dialog appears for each program selected. Here, you have to confirm the software installation folder. The matching component is then copied into this folder. The individual components include documents describing how the components work and how they are set up. You can find these documents in the folder specified after the installation.

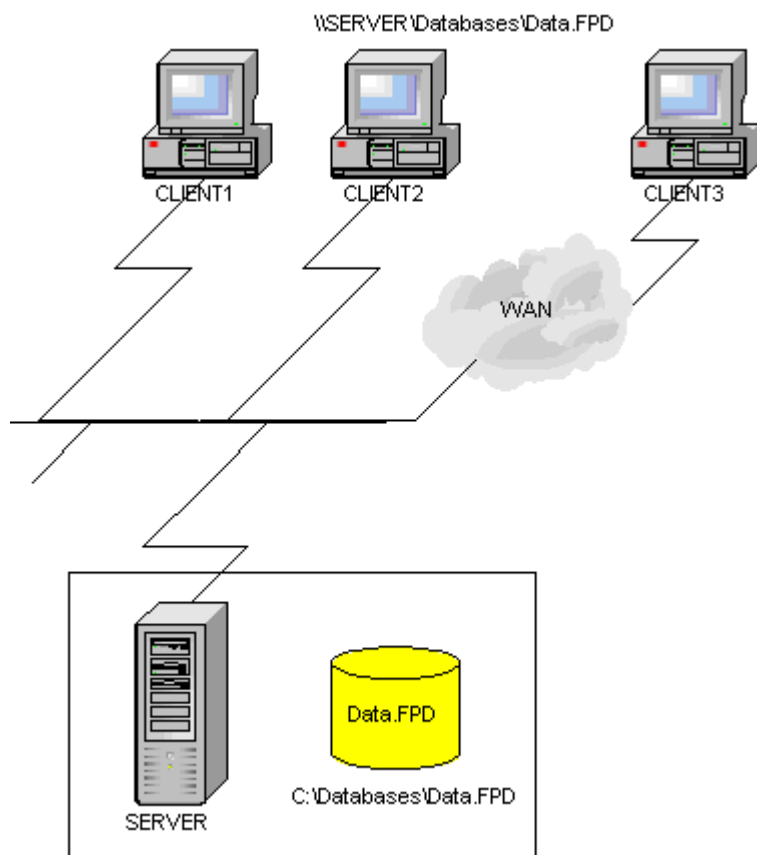
### DCOM

DCOM stands for Distributed Component Object Model. DCOM is a protocol that enables software components to communicate with each other reliably, securely and efficiently over a network.

DCOM is part of Windows 2000/XP/Vista and is installed during the installation of the operating system.

FlexPro requires DCOM if you wish to use the FPAccess interface or one of the other supported interfaces for data acquisition software based on it.

A typical DCOM application could therefore look like this:



FlexPro runs on one server and has a project database (C:\Databases\Data.fpd) open. One or more clients for their part open the same project database through a shared network folder (\\SERVER\Databases\Data.fpd) to write acquired data. The user on the server computer sees how the data is written into the project database and can immediately begin working with the data.

For DCOM to be used, it must first be configured. FlexPro must also be configured to use DCOM.

After following the instructions, configure DCOM on the computer that has FlexPro installed on it.

1. Click on **Start**.
2. In the text box **Start Search** enter **dcomcnfg** and press ENTER to start DCOM configuration.
3. In the **User Account Control** dialog box that appears, click on **Continue**.
4. Open the **Component Services** node and then **Computers**.
5. Right-click with your mouse on **My Computer** and then select **Properties** from the context menu.
6. Switch to the **Default Properties** tab and make sure that **Enable Distributed COM on this computer** is selected.
7. Click on **OK**, to close the dialog box.
8. Open the **My Computer** node and then **DCOM Configuration**.
9. Choose the **FlexPro Project Database** entry and then display its properties.
10. Switch to the **General** tab and choose the entry **None** as the **Authentication Level**.
11. Switch to the **Security** tab and make sure that the user who needs access to FlexPro from a remote system has the appropriate permissions to do so. To do this, for the launch and access permissions, select the **Customize** option, click on **Edit** and add a suitable user group.
12. Switch to the **Identity** tab. Select either **Interaktiver Benutzer** or a specific user whose user account is to be used for running the application. If you select a specific user, you must specify this user's password.
13. Note: To test the configuration, it makes sense to set the identity to **The interactive user** first, since you will not be able to access the running FlexPro program otherwise.
14. Click on **OK**, and then on the **File** menu, click on **Exit** to exit the configuration.

After following the instructions, configure DCOM on the computer that has FlexPro installed on it.

1. Click on **Start** and then on **Run**.
2. Enter **dcomcnfg** and then click on **OK** to start DOM configuration.
3. Open the **Component Services** node and then **Computers**.

4. Right-click with your mouse on **My Computer** and then select **Properties** from the context menu.
5. Switch to the **Default Properties** tab and make sure that **Enable Distributed COM on this computer** is selected.
6. Click on **OK** to close the dialog box
7. Open the **My Computer** node and then **DCOM Configuration**.
8. Choose the **FlexPro Project Database** entry and then display its properties.
9. Switch to the **General** tab and choose the entry **None** for the **Authentication Level**.
10. Switch to the **Security** tab and make sure that the user who needs access to FlexPro from a remote system has the appropriate permissions to do so. To do this, for the launch and access permissions, select the **Customize** option, click on **Edit** and add a suitable user group.
11. Switch to the **Identity** tab. Select either **The interactive user** or a specific user whose user account is to be used for running the application. If you select a specific user, you must specify this user's password.
12. Note: To test the configuration, it makes sense to first set the identity to **The interactive user**, since you will not be able to access the running FlexPro program otherwise.
13. Click on **OK** and then on **Exit** in the **File** menu to exit the configuration.

After following the instructions, configure DCOM on the computer that has FlexPro installed on it.

1. Click on **Start** and then on **Run**.
2. Enter **dcomcnfg** and then click on **OK** to start DOM configuration.
3. Switch to the **Default Properties** tab and make sure that **Enable Distributed COM on this computer** is selected.
4. On the **Applications** tab, choose the **FlexPro Project Database** entry and then display its properties.
5. Switch to the **General** tab and choose the entry **(none)** for the **Authentication Level**.
6. Switch to the **Security** tab and make sure that the user who needs access to FlexPro from a remote system has the appropriate permissions to do so. To do this, add a suitable user group to the launch and access permissions (in case of doubt, choose

Any).

7. Switch to the **Identity** tab. Select either **The interactive user** or a specific user whose user account is to be used for running the application. If you select a specific user, you must specify this user's password.
  8. **Note:** To test the configuration, it makes sense to first set the identity to **The interactive user**, since you will not be able to access the running FlexPro program otherwise.
  9. Click **OK** three times to exit the configuration.
- 
- FlexPro does not have to be launched manually on the server computer. Windows launches FlexPro automatically in the background, provided FlexPro is used as the project database server.
  - If you are on the client computer and want to access the project database C:\Project Databases\Data.fpd that is stored on the server computer called SERVER, you cannot use a network shared root (whole drive) to access the file. In this case, share the directory C:\Project Databases on SERVER and access it from the client via \\SERVER\Project Databases\Data.fpd. If you have set up the whole drive C: to be shared and want to access the project database with \\SERVER\C\Project Databases\Data.fpd, it will cause an access violation. This is a known issue.

## Remote Control of FlexPro via DDE

### DDE Interface

DDE (**D**ynamic **D**ata **E**xchange) is a dynamic exchange of data between Windows applications. A client application that initiates the exchange is required. A server application is also required to serve the requested information or run the requested actions. FlexPro can function as a server for DDE communication and can run commands that are sent from the client. A typical client would be, for instance, a Visual Basic application that accepts data from a device, saves the information in a file and remotely controls FlexPro for the evaluation and documentation of the data.

---

**Note:** The latest FlexPro Automation interface offers several more comprehensive options for communication between FlexPro and a client application. It is recommended that you use the DDE interface.

---



## Connection Setup, Interaction and Termination

To be able to run FlexPro functions from other Windows programs that support DDE communication as a client, you first have to set up a DDE connection. This is achieved by sending the message WM\_DDE\_INITIATE, where "FlexPro" or "FlexPro 10" is specified as the application and "SYSTEM" is specified as the topic. If a connection is successfully made, there is a acknowledgement by way of a WM\_DDE\_ACK message. If you have provided a version number, the connection will only be successful when this particular version of FlexPro is installed.

Any number of commands can then be sent to FlexPro to be run with the help of the message WM\_DDE\_EXECUTE. It is also possible to send several commands together with a single message. When a command is correctly received, this is confirmed by a WM\_DDE\_ACK message. There is also a confirmation if the commands received cannot be run correctly.

When the client application has sent all of the desired commands to FlexPro, the DDE communication is ended by sending the WM\_DDE\_TERMINATE message.

## Visual Basic Example

Here is a brief example of how DDE communication used for printing a document can be achieved by means of a client application in Visual Basic:

```
Sub Document_Click ()
    Const NONE = 0, LINK_MANUAL = 2          ' Constants
    Dim Cmd, Z                                ' Variables

    ' Combine commands:
    Cmd = "[open(""c:\Project Databases\Test.fpd""])"
    Cmd = Cmd & "[openobject(""Document.doc""])"
    Cmd = Cmd & "[update]"
    Cmd = Cmd & "[print]"
    Cmd = Cmd & "[closeobjects]"
    Cmd = Cmd & "[close]"

    If Text1.LinkMode = NONE Then
        Z = Shell("FlexPro", 4)              ' Start FlexPro
        Text1.LinkTopic = "FlexPro 10|System" ' Set LinkTopic
        Text1.LinkItem = ""                  ' LinkItem
```

```
Text1.LinkMode = LINK_MANUAL      ' LinkMode
End If
Text1.LinkExecute Cmd              ' Execute Commands
End Sub
```

This subroutine can be called, for instance, if the source data on which the document is based has been overwritten by a data acquisition program.

With the commands available, it is possible to open and close project databases, to open, update, print and close an object in the project database, as well as to activate folders.

The commands listed here can, as described above, be sent to FlexPro with a WM\_DDE\_EXECUTE message, individually or bundled.

---

**Note:** In FlexPro it is basically possible to work with several project databases simultaneously. This is not possible in connection with DDE communication, since the project database referred to would have to be specified with every command, except when opening a project database.

---

## **9 End User License Agreement**

### **9.1 FlexPro End User License Agreement (DE AT CH)**

#### **1. Scope**

The following provisions apply to any end user license agreement entered into with users (hereinafter referred to as "Licensee") headquartered within Germany, Austria and Switzerland.

#### **2. Subject of the agreement**

The subject of this agreement is the computer software FlexPro developed by the company Weisang GmbH (hereinafter referred to as "Weisang"), the software description and user manual as well as all other related written material, collectively referred to hereinafter as "software". Weisang makes this software available to the Licensee as per the material scope of use and time period agreed upon with Weisang.

As agreed, the Licensee will receive the software either recorded to a data carrier or USB stick or as a download via download link or as an e-mail attachment. The is not entitled to claim a disclosure of the source code.

The software manufacturer Weisang draws attention to the fact that despite applying state-of-the-art programming technology, it is not possible to write software that will work correctly under all circumstances. The subject of this agreement is only a software product which can be used as intended based on its description and as specified in the user manual.

This software has not been designed for a use in hazardous environments. Consequently, it is not suited for security or safety related applications in nuclear facilities, in aircraft GPS and communications systems, for air traffic control, weapons systems, or life-support machines.

#### **3. Scope of use**

The right of use (hereinafter referred to as "license") for the FlexPro software is limited as follows:

##### **a) 30-day evaluation license**

Weisang grants the Licensee the ordinary, non-exclusive and personal right to use the FlexPro software on a single computer at a single location for a period of 30 days from the date of its first use. This use shall be limited to evaluation purposes. In particular the Licensee shall not publish or otherwise use the evaluation and Visual Basic programs created using this FlexPro license prior to acquiring a single or network license for

FlexPro.

### **b) Reader license**

Weisang grants the Licensee the ordinary, non-exclusive and personal right to install and use the FlexPro software on a single desktop PC and a portable computer (notebook). This use is restricted to read-only access to analyses, which have been created using a FlexPro single user license or a network license. In particular, the Licensee may not use this FlexPro license to create analyses or as a runtime environment for automated analyses.

### **c) Single user license**

Weisang grants the Licensee the ordinary, non-exclusive and personal right to use the FlexPro software on a single desktop PC at a single location. The Licensee may also install the software to a portable computer (notebook), provided that the Licensee is the only user of this portable computer and that the FlexPro software will always only be used on a single computer.

### **d) Network license**

Weisang grants the Licensee the right to install the FlexPro software to any number of workstation computers located in the same local area network (LAN) on which the FlexPro network license is installed. The distance between the workstation computers on which the network license is installed and the associated server must not exceed eight kilometers. The number of concurrent FlexPro users must never exceed the number of workstations for which a network license is granted.

### **e) Runtime license**

Weisang grants the Licensee the ordinary, non-exclusive and personal right to use the FlexPro software on a single computer as the runtime environment for automated analyses, which have been created using a FlexPro single user or network license. Any interactive use of the runtime license is prohibited. Any operation of the runtime license for server applications shall be subject to Weisang's written approval. To this end, Weisang will charge additional license fees, the amount of which will depend on the number of individuals using the server.

Any further use is prohibited. Depending on the granted license, the following additional limitations apply:

### **a) University license**

Use at schools and universities shall be made for teaching and non-commercial research purposes only. Thus, use in projects financed by the industry or other principals is prohibited.

### **b) Student license**

Use is only permitted for registered pupils and students and restricted exclusively to

activities related to personal continued education. Consequently, any use in research and professional environments for commercial or industrial purposes is prohibited.

**c) License for non-profit organizations**

This license may be used by members of non-profit organizations. Any use for commercial or industrial purposes is prohibited.

**d) Subscription**

The right to use the FlexPro software is limited to one year and includes maintenance carried out during this period. Maintenance includes the following services provided by Weisang:

- Free e-mail or telephone product support
- Migration of a license to another computer
- Free update to the next release number, to the extent that maintenance for the license is continuously maintained. If the maintenance services for an existing license are acquired at a later date, any update to the next major release number shall only be possible after a period of one year.

Weisang will submit to licensee an offer to extend the subscription by another period of one year 60 days before the subscription expires. If this offer is accepted and the related invoice paid, this agreement shall be extended by another year starting when the previous subscription expires.

#### **4. Copyrights, use and processing rights**

(1) The software provided by Weisang is copyright protected. This software is a computer program capable of being protected by copyright as specified under Art. 2, Sect. 1, No. 1, Para. 69 a of the UrhG [German copyright law]. In the relationship between the parties, Weisang is sole owner of any copyright to the software and to any documentation provided during the negotiations leading to the conclusion and execution of this agreement.

(2) The Licensee is granted an ordinary, non-exclusive right, which is unrestricted as to time and place (hereinafter referred to as "license"), to use the software as specified in this agreement and in the associated documentation. Contractual use means that the Licensee shall be authorized to install the software in accordance with Subsection 2 of this agreement and to save it to the RAM and hard disks of the hardware used.

(3) Reproduction of the software shall be permitted insofar as this is necessary for backup purposes (backup copy) only. This backup copy shall be identified as such and (if technically feasible) shall bear the copyright notice of the original data carrier. The software and the written materials must not be combined with any other software, and shall not be copied or duplicated, in full or in part, or in their original or modified form.

(4) It is prohibited to modify or deface the copyright notices, further legal reservations or other characteristics serving to identify the programming contained in the software.

(5) The rights of use will only be granted definitively to the Licensee once the agreed fees have been paid in full. Until payment has been received in full, the Licensee shall be authorized to use the software until notice to the contrary.

## **5. Special restrictions**

The Licensee shall not modify, translate or create works derived from the software without prior written approval of Weisang. The Licensee shall not reverse engineer, decompile, or disassemble the software, or in any way attempt to make its source code accessible. Any claim to disclosure of the source code is excluded.

## **6. Transfer of the right of use**

The right to use the software may only be transferred to third parties upon prior written approval from Weisang and only under the terms of this agreement. Lending or leasing of the software is explicitly prohibited.

## **7. Duration**

The term of this agreement depends on the scope of use agreed to between the parties (Subsection 2 of this agreement). The Licensee's right to use the software shall expire in the event that the Licensee substantially violates any of the provisions herein and fails to end this breach of contract even after receipt of a notice by Weisang granting a period to end such breach. In such a case the Licensee will no longer be authorized to continue using the software and shall be obliged to uninstall and definitively delete the software. Upon termination of this right of use, the Licensee shall destroy all original data carriers and any copies of the software including any modified versions and related written materials; and upon Weisang's request, the Licensee shall declare its complete destruction by way of an affidavit.

## **8. Indemnity in case of a breach of this agreement**

Weisang draws attention to the Licensee that Licensee shall be liable for any damage resulting from copyright infringement which affects Weisang as a result of the Licensee's breach of the contractual terms.

## **9. Infringement of property rights of third parties**

(1) In the event that third parties raise claims against the Licensee due to an infringement of their property rights through the software, and if the use of the software is adversely affected or prohibited by such claim, Weisang will be liable as follows:

At the Licensee's election and costs Weisang will either modify or replace the software so that the property right will no longer be infringed or hold the Licensee harmless against license fees requested by the owner of the property right or a third party.

If Weisang does not succeed in doing this under reasonable conditions, Weisang will notify the Licensee and prohibit the use of the Software in whole or in part (in the case of a module) from a specified time going forward. Weisang will then indemnify the Licensee by refunding the purchase price for the license concerned and/or the license for the affected module. Rights to further indemnities shall only exist as provided under Subsection 10, Paragraphs (3) and (4).

Weisang may only be held liable if the Licensee notifies Weisang promptly on of any claim raised by a third party, if Weisang does not acknowledge the alleged property right violation and transfers any form of dispute including out-of-court settlements to Weisang or only leads them after consultation with Weisang. Licensee's necessary court costs and reasonable attorneys' fees for the defense of such claim and incurred after such consultation with Weisang shall be borne by Weisang.

(2) In the event that Licensee discontinues using the software in order to mitigate possible damages or for another good cause, the Licensee shall be obliged to notify such third party that this end of usage does not imply acknowledgement of the alleged infringement of property rights.

(3) Claims against Weisang shall be excluded to the extent that the Licensee is responsible for such infringement of property rights.

(4) Further rights arising from an infringement of the property rights of third parties shall only exist as provided under Subsection 10, Paragraphs (3) and (4).

## **10. Modifications and updates**

Weisang is entitled to develop software updates at its own discretion. Weisang is not obligated to make updates of this software available to Licensees who have not entered into a maintenance agreement or have not acquired an update.

## **11. Warranty and liability of the manufacturer**

(1) Weisang provides to the Licensee the software free of material defects and defects of title. Functional impairments resulting from the hardware and software environment provided by the Licensee or from incorrect operation, external damaging data, and faults in computer networks or other reasons related to Licensee's inherent risks shall not be deemed a defect. Weisang may not be held liable for software which has been modified by the Licensee unless the Licensee proves that such modification is not related to the notified defect.

(2) In the event of defects, the Licensee shall have the legal warranty rights as provided

for hereinafter.

In the event of material defects, Weisang, at its discretion, shall fulfill the warranty obligation either through remedy or replacement. Subsequent fulfillment may also be carried out particularly through the provision of a new program version. In case that subsequent fulfillment fails, the Licensee shall be entitled to withdraw from the agreement.

(3) Weisang shall be liable for damages only under the following circumstances: The damage is the result of intent or gross negligence on the part of Weisang; in this case, Weisang shall be liable without limitation. Furthermore, Weisang shall be liable without limitation in the case of grossly negligent breach in the performance of its fundamental contractual obligations. Weisang shall also be liable for gross negligence of fundamental contractual obligations; in this case, however, liability on the part of Weisang is limited to foreseeable damages that are typical for this type of contract.

(4) Nevertheless, in the case of ordinary negligence, Weisang shall not be liable for consequential damage caused by defects and lost profits.

(5) The aforesaid limitations of liability do not apply to loss of life, physical injury and damage to health, for a defect after assumption of a guarantee for the quality of the contractual product and in the event of maliciously concealed defects. Liability pursuant to the German Product Liability Act (Produkthaftungsgesetz) shall remain unaffected.

## **12. Applicable law and place of jurisdiction**

(1) Any and all legal relations under the present contract shall be governed by the law of the Federal Republic of Germany. Application of the UN Convention on Contracts for the International Sale of Goods (C.I.S.G.) is excluded. The contract language is German.

(2) Place of jurisdiction for every dispute, irrespective of its legal grounds, and arising in connection with this agreement, shall be St. Ingbert, Germany, to the extent that licensee is a dealer, a legal person of public law or special fund under public law or equivalent to that or when its place of business or subsidiary is located abroad. Should there be discrepancies between the interpretation of the German and English versions of this End User License Agreement, the German version shall prevail.

As of 22 December 2014



## **9.2 FlexPro End User License Agreement (International)**

### **1. Scope**

The following provisions apply to any end user license agreement entered into with users (hereinafter referred to as "Licensee") headquartered outside of Germany, Austria and Switzerland.

### **2. Subject of the agreement**

The subject of this agreement is the computer software FlexPro developed by the company Weisang GmbH (hereinafter referred to as "Weisang"), the software description and user manual as well as all other related written material, collectively referred to hereinafter as "software". Weisang makes this software available to the Licensee as per the material scope of use and time period agreed upon with Weisang.

As agreed, the Licensee will receive the software either recorded to a data carrier or USB stick or as a download via download link or as an e-mail attachment. The is not entitled to claim a disclosure of the source code.

The software manufacturer Weisang draws attention to the fact that despite applying state-of-the-art programming technology, it is not possible to write software that will work correctly under all circumstances. The subject of this agreement is only a software product which can be used as intended based on its description and as specified in the user manual.

This software has not been designed for a use in hazardous environments. Consequently, it is not suited for security or safety related applications in nuclear facilities, in aircraft GPS and communications systems, for air traffic control, weapons systems, or life-support machines.

### **3. Scope of use**

The right of use (hereinafter referred to as "license") for the FlexPro software is limited as follows:

#### **a) 30-day evaluation license**

Weisang grants the Licensee the ordinary, non-exclusive and personal right to use the FlexPro software on a single computer at a single location for a period of 30 days from the date of its first use. This use shall be limited to evaluation purposes. In particular the Licensee shall not publish or otherwise use the evaluation and Visual Basic programs created using this FlexPro license prior to acquiring a single or network license for FlexPro.

**b) Reader license**

Weisang grants the Licensee the ordinary, non-exclusive and personal right to install and use the FlexPro software on a single desktop PC and a portable computer (notebook). This use is restricted to read-only access to analyses, which have been created using a FlexPro single user license or a network license. In particular, the Licensee may not use this FlexPro license to create analyses or as a runtime environment for automated analyses.

**c) Single user license**

Weisang grants the Licensee the ordinary, non-exclusive and personal right to use the FlexPro software on a single desktop PC at a single location. The Licensee may also install the software to a portable computer (notebook), provided that the Licensee is the only user of this portable computer and that the FlexPro software will always only be used on a single computer.

**d) Network license**

Weisang grants the Licensee the right to install the FlexPro software to any number of workstation computers located in the same local area network (LAN) on which the FlexPro network license is installed. The distance between the workstation computers on which the network license is installed and the associated server must not exceed eight kilometers. The number of concurrent FlexPro users must never exceed the number of workstations for which a network license is granted.

**e) Runtime license**

Weisang grants the Licensee the ordinary, non-exclusive and personal right to use the FlexPro software on a single computer as the runtime environment for automated analyses, which have been created using a FlexPro single user or network license. Any interactive use of the runtime license is prohibited. Any operation of the runtime license for server applications shall be subject to Weisang's written approval. To this end, Weisang will charge additional license fees, the amount of which will depend on the number of individuals using the server.

Any further use is prohibited. Depending on the granted license, the following additional limitations apply:

**a) University license**

Use at schools and universities shall be made for teaching and non-commercial research purposes only. Thus, use in projects financed by the industry or other principals is prohibited.

**b) Student license**

Use is only permitted for registered pupils and students and restricted exclusively to activities related to personal continued education. Consequently, any use in research and professional environments for commercial or industrial purposes is prohibited.

**c) License for non-profit organizations**

This license may be used by members of non-profit organizations. Any use for commercial or industrial purposes is prohibited.

**d) Subscription**

The right to use the FlexPro software is limited to one year and includes maintenance carried out during this period. Maintenance includes the following services provided by Weisang:

- Free e-mail or telephone product support
- Migration of a license to another computer
- Free update to the next release number, to the extent that maintenance for the license is continuously maintained. If the maintenance services for an existing license are acquired at a later date, any update to the next major release number shall only be possible after a period of one year.

Weisang will submit to licensee an offer to extend the subscription by another period of one year 60 days before the subscription expires. If this offer is accepted and the related invoice paid, this agreement shall be extended by another year starting when the previous subscription expires.

**4. Copyrights, use and processing rights**

(1) The software provided by Weisang is copyright protected. This software is a computer program capable of being protected by copyright as specified under Art. 2, Sect. 1, No. 1, Para. 69 a of the UrhG [German copyright law]. In the relationship between the parties, Weisang is sole owner of any copyright to the software and to any documentation provided during the negotiations leading to the conclusion and execution of this agreement.

(2) The Licensee is granted an ordinary, non-exclusive right, which is unrestricted as to time and place (hereinafter referred to as "license"), to use the software as specified in this agreement and in the associated documentation. Contractual use means that the Licensee shall be authorized to install the software in accordance with Subsection 2 of this agreement and to save it to the RAM and hard disks of the hardware used.

(3) Reproduction of the software shall be permitted insofar as this is necessary for backup purposes (backup copy) only. This backup copy shall be identified as such and (if technically feasible) shall bear the copyright notice of the original data carrier. The software and the written materials must not be combined with any other software, and shall not be copied or duplicated, in full or in part, or in their original or modified form.

(4) It is prohibited to modify or deface the copyright notices, further legal reservations or other characteristics serving to identify the programming contained in the software.

(5) The rights of use will only be granted definitively to the Licensee once the agreed fees have been paid in full. Until payment has been received in full, the Licensee shall be authorized to use the software until notice to the contrary.

## **5. Special restrictions**

The Licensee shall not modify, translate or create works derived from the software without prior written approval of Weisang. The Licensee shall not reverse engineer, decompile, or disassemble the software, or in any way attempt to make its source code accessible. Any claim to disclosure of the source code is excluded.

## **6. Transfer of the right of use**

The right to use the software may only be transferred to third parties upon prior written approval from Weisang and only under the terms of this agreement. Lending or leasing of the software is explicitly prohibited.

## **7. Duration**

The term of this agreement depends on the scope of use agreed to between the parties (Subsection 2 of this agreement). The Licensee's right to use the software shall expire in the event that the Licensee substantially violates any of the provisions herein and fails to end this breach of contract even after receipt of a notice by Weisang granting a period to end such breach. In such a case the Licensee will no longer be authorized to continue using the software and shall be obliged to uninstall and definitively delete the software. Upon termination of this right of use, the Licensee shall destroy all original data carriers and any copies of the software including any modified versions and related written materials; and upon Weisang's request, the Licensee shall declare its complete destruction by way of an affidavit.

## **8. Indemnity in case of a breach of this agreement**

Weisang draws attention to the Licensee that Licensee shall be liable for any damage resulting from copyright infringement which affects Weisang as a result of the Licensee's breach of the contractual terms.

## **9. Infringement of property rights of third parties**

(1) In the event that third parties raise claims against the Licensee due to an infringement of their property rights through the software, and if the use of the software is adversely affected or prohibited by such claim, Weisang will be liable as follows:

At the Licensee's election and costs Weisang will either modify or replace the software so that the property right will no longer be infringed or hold the Licensee harmless against

license fees requested by the owner of the property right or a third party.

If Weisang does not succeed in doing this under reasonable conditions, Weisang will notify the Licensee and prohibit the use of the Software in whole or in part (in the case of a module) from a specified time going forward. Weisang will then indemnify the Licensee by refunding the purchase price for the license concerned and/or the license for the affected module. Rights to further indemnities shall only exist as provided under Subsection 10, Paragraphs (3) and (4).

Weisang may only be held liable if the Licensee notifies Weisang promptly on of any claim raised by a third party, if Weisang does not acknowledge the alleged property right violation and transfers any form of dispute including out-of-court settlements to Weisang or only leads them after consultation with Weisang. Licensee's necessary court costs and reasonable attorneys' fees for the defense of such claim and incurred after such consultation with Weisang shall be borne by Weisang.

(2) In the event that Licensee discontinues using the software in order to mitigate possible damages or for another good cause, the Licensee shall be obliged to notify such third party that this end of usage does not imply acknowledgement of the alleged infringement of property rights.

(3) Claims against Weisang shall be excluded to the extent that the Licensee is responsible for such infringement of property rights.

(4) Further rights arising from an infringement of the property rights of third parties shall only exist as provided under Subsection 10, Paragraphs (3) and (4).

## **10. Modifications and updates**

Weisang is entitled to develop software updates at its own discretion. Weisang is not obligated to make updates of this software available to Licensees who have not entered into a maintenance agreement or have not acquired an update.

## **11. Warranty**

(1) Weisang provides to the Licensee the software free of material defects and defects of title. Functional impairments resulting from the hardware and software environment provided by the Licensee or from incorrect operation, external damaging data, and faults in computer networks or other reasons related to Licensee's inherent risks shall not be deemed a defect. Weisang may not be held liable for software which has been modified by the Licensee unless the Licensee proves that such modification is not related to the notified defect.

(2) In the event of defects, the Licensee shall have the legal warranty rights as provided for hereinafter.

In the event of material defects, Weisang , at its discretion, shall fulfill the warranty

obligation either through remedy or replacement. Subsequent fulfillment may also be carried out particularly through the provision of a new program version. In case that subsequent fulfillment fails, the Licensee shall be entitled to withdraw from the agreement.

### **12. Liability**

(1) Weisang shall only be liable for damages, in particular consequential damages and lost profit, to the extent that Weisang is liable under the terms of the law applicable to the Licensee.

### **13. Place of jurisdiction**

Place of jurisdiction for every dispute, irrespective of its legal grounds, and arising in connection with this agreement, shall be St. Ingbert, Germany, to the extent that licensee is a dealer, a legal person of public law or special fund under public law or equivalent to that or when its place of business or subsidiary is located abroad. Should there be discrepancies between the interpretation of the German and English versions of this End User License Agreement, the German version shall prevail.

As of 22 December 2014

# Index

## - " -

"Natural Neighbor" Interpolation 305

## - 2 -

### 2D Diagram

- (Description) 297
- Adding a Boundary Line 189
- Adding a Dimension Line 189
- Adding a Perpendicular 189
- Adding a Symbol 189
- Adding a Tangent 189
- Adding a Tolerance Band 189
- Adding Curve 330
- Adding or Removing an Axis 320
- Axis 298
- Changing Page Splitting 318
- Changing the Background Color 317
- Color Palette and Color Gradient 314
- Creating 315
- Cursor Environment 161
- Designing Grid Elements 317
- Dimensioning a Curve 189
- Display Formats 303
- Displaying in the Coordinates Window 188
- Displaying the Curve Data 184
- Embedding Graphics 319
- Enabling or Disabling Automatic Margin Correction 316
- Exporting 387
- Grid 315
- Legend 309
- Multi-page 294

- Opening a Curve Data Set 331
- Polar Graph 312
- Resizing 316
- Showing a Page 318
- Specifying Output Unit for Axis 320
- Template 294
- Toggling Grid Lines 317
- Translating a Curve 189, 190

## - 3 -

### 3D Diagram

- (Description) 297
- Adding a Boundary Line 189
- Adding a Data Series 332
- Adding a Dimension Line 189
- Adding a Perpendicular 189
- Adding a Symbol 189
- Adding a Tangent 189
- Adding a Tolerance Band 189
- Axis 298
- Changing Lighting Parameters 319
- Changing the Background Color 317
- Changing the Viewing Angle 318
- Color Legend 315
- Color Palette and Color Gradient 314
- Creating 315
- Cursor Environment 161
- Curve Types and Display Formats 305
- Designing Grid Elements 317
- Dimensioning a Curve 189
- Displaying in the Coordinates Window 188
- Displaying the Curve Data 184
- Embedding Graphics 319
- Enabling or Disabling Automatic Margin Correction 316
- Exporting 387
- Floor 298
- Grid 315
- Legend 309
- Lighting 310

## 3D Diagram

- Opening a Curve Data Set 331
- Resizing 316
- Specifying Output Unit for Axis 320
- Toggling Grid Lines 317
- View 310

## 3D View 310

**- A -**

## Absolute Path Name 219

## Absolute Time 98

- AbsoluteTime Operator (FPScript) 254
- AbsoluteTimeStrict Operator (FPScript) 254

## AbsoluteTime Operator (FPScript) 254

## AbsoluteTimeStrict Operator (FPScript) 254

## Accessing Data (FPScript) 219

## Accessing Header Information (FPScript) 239

## Accessing Object Properties (FPScript) 218

## Acquisition Programs, Data Transfer 499

## ActivatedFolder 219

## Activating or Deactivating the Magnetic Grid 373

## Activation

- Offline 15
- Online 15

## Adapting

- Number of Fractional Digits for Floating Point Values 426
- Output Format for Complex Numbers 426

## Analysis

- (Description) 191
- Creating Templates 198

## Efficiency 197

## Multiple Measurements 196

## Organizing Templates 199

## Structuring 194

## Template (Description) 191

## Using Templates 198

## Analysis Object

- (Description) 244
- Converting into a Data Set 109
- Converting into a Formula 109
- Creating 245
- Editing 246
- Editing Header Information 108
- Exporting 148
- Exporting as a File 153
- Exporting via OLE 153
- Exporting via the Clipboard 154
- Playing as Audio File 107

## Analysis Template 191

- Creating 198
- Using 198

## Analysis Templates

- Organizing 199

## Angle

- Formatting 393

## Animate

- Cursor 161, 170

## Application Profile 426

## Argument

- from FPScript Functions 241

## Arithmetical Expressions 208

## As Object Keyword (FPScript) 262

## ASAM ODS

- Importing Data (Description) 138

## Assigned Data Object 98

## Associativity 208

## Associativity of Operators (FPScript) 234

## Audio File

- as Presentation Object 364



- 
- Audio Playback
    - Changing Speed 368
  - Auto Macro 471
  - Automation
    - Analysis 32
    - Communicating with Other Applications 463
    - Object Hierarchy 453, 455
    - Object Model 452
    - Using Macros 428
  - Axis
    - (Description) 298
    - Adding a Unit to a Division Label 325
    - Adding or Deleting a Division 322
    - Adding or Deleting a Division Label 323
    - Adding or Deleting a Label 327
    - Adding or Deleting an Orientation Arrow 325
    - Adding or Removing in a 2D Diagram 320
    - Adjusting the Height of Stacked Y Axes 321
    - Assigning to Curve 330
    - Changing Division Length and Type 322
    - Changing Label Alignment 328
    - Changing Label Position 328
    - Changing the Number of Subdivisions 327
    - Changing the Origin 320
    - Changing the Scaling 327
    - Customizing Scaling 326
    - Displaying an 3D Axis System as an Axis Intercept 328
    - Displaying Data Set Information in the Label 328
    - Division Labeling via Data Set 298
    - Division Labeling via Function 298
    - Formatting the Division Label 324
    - Isometric Adjustment 298
    - Labeling 298, 310
    - Labeling Divisions with Text 324
    - Labeling Only the End Values 328
    - Metric Scaling 298
    - Positioning in a 2D Diagram 321
    - Positioning Side By Side in a 2D Diagram 322
    - Positioning the Division Label 324
    - Scaling Types 298
    - Specifying the Output Unit 320
    - Stacking in a 2D Diagram 321
    - Tagging Grid Lines 317
    - Tagging Isometric Adjustment On and Off 323
  - Axis Division
    - (Description) 298
    - Adding or Deleting 322
    - Labeling with a Unit 325
    - Labeling with Text 324
  - Axis Intercept (3D Diagram) 328
  - Axis Label
    - Adding or Deleting 327
    - Changing Alignment 328
    - Changing Position 328
    - Displaying Data Set Information 328
    - Writing Directly on Axis 327
  - Axis Labeling
    - (Description) 298
  - Axis Origin
    - (Description) 298
    - Changing 320
    - Positioning Axis in 321
  - Axis Scaling
    - Changing 327
    - Customizing 326
    - Types 298
- B -**
- Bars
    - 2D Curve 303
    - Changing the Border 339
    - Changing the Bottom 335
-

**Bars**

- Changing Width 334
- Designing 334

**Base Units (SI) 127****Binary Data**

- Importing 141

**Boolean Value (FPScript) 98****Border**

- Changing for Columns 339

**Bound Cursor 161****Boundary Line**

- Adding to a Diagram 189

**Breakpoint**

- (Description) 204
- Setting and Removing 206

**Button**

- Adding 404
- Adding Image 417
- Adding Text 417
- Changing Image 414
- Copying 405
- Deleting 405
- Grouping on Toolbar 408
- Moving 405
- Renaming 410
- Resizing 413
- Showing All on Toolbar 412

**- C -****Calendar Time**

- Formatting 393

**Calendar Time (FPScript) 98****Canceling (Checkout) 17****Canceling Checkout 17****Category**

- Assigning Objects 92
- Displaying in the Object List 82

**Cell**

- Editing 117
- Tables 347

**Cell Table**

- (Description) 347
- Adding a Column 357
- Adding a Row 357
- Adding or Deleting a Label 351
- Adding or Deleting Dividing Lines and Frames 350
- Changing Colors 350
- Changing the Column Width 358
- Changing the Font 350
- Changing the Number Format 360
- Changing the Row Height 359
- Changing the Text Alignment 351
- Creating 349
- Deleting Rows or Columns 358
- Displaying Data 359
- Displaying in the Coordinates Window 188
- Editing Text 359
- Embedding Graphics 351
- Exporting 387
- Positioning the Label 351
- Rearranging Columns or Rows 358
- Specifying the Output 358

**Changing a Fill Color 381****Changing a Pattern 381****Changing Cycle Duration 92****Changing the Background Color 317****Changing the Drawing Grid 373****Check Out 16****Color Gradient 314****Color Legend**

- (Description) 315
- Adding or Deleting a Division 322
- Adding or Deleting a Division Label 323
- Adding or Deleting an Orientation Arrow 325

- 
- Color Legend
    - Changing Division Length and Type 322
    - Formatting the Division Label 324
    - Positioning 318
    - Positioning the Division Label 324
  - Color Palette 314
    - Determining the Interval Limits for a Curve 341
    - Editing for a Curve 342
    - Transferring 342
    - Turning On 340
  - Color Ramp
    - for Curve 336
  - Column
    - Calculating Obtained Values 122
    - Changing the Quantity in a Legend 346
    - Filling with Values 122
  - Column Index 347
  - Column Table
    - (Description) 347
    - Adding a Column Title 355
    - Adding a Data Column 352
    - Adding an Index Column 352
    - Adding or Deleting a Column Index 356
    - Adding or Deleting a Label 351
    - Adding or Deleting Dividing Lines and Frames 350
    - Alternating Colors 354
    - Changing Colors 350
    - Changing the Column Width 353
    - Changing the Font 350
    - Changing the Number Format 354
    - Changing the Row Height 353
    - Changing the Text Alignment 351
    - Creating 349
    - Displaying in the Coordinates Window 188
    - Embedding Graphics 351
    - Exporting 387
    - Merging Cells 360
    - Multi-Page 294, 347
    - Outputting Data with a Unit Symbol 355
    - Positioning the Label 351
    - Rearranging Columns 353
    - Show Page 354
    - Specifying the Output Unit 352
    - Template 294
  - Column Title
    - Displaying Data Set Information 356
  - Columns
    - 2D Curve 303
    - 3D Curve 305
    - Adding a Data Series 332
    - Changing Fill Color and Style 337
    - Changing the Border 339
    - Changing the Bottom 335
    - Changing Width 334
    - Color Ramp 336
    - Designing 334
    - Determining the Interval Limits for a Color Palette 341
    - Editing a Color Palette 342
    - Rearranging Rows 329
    - Turning On a Color Palette 340
  - Combining
    - FPScript Statements 222
  - Command
    - Adding 404
    - Adding Image 417
    - Adding Text 417
    - Deleting 405
    - Moving 405
    - Renaming 410
  - Command Bar
    - (Description) 473
    - Using 474
  - Command Line Parameters for FlexPro 424
  - Comments (FPScript) 222
  - Communicating with Other Applications 463
-

- Compatibility 50
- Complex Number (FPScript) 98
- Complex Numbers
  - Changing Output Format 426
- Component (FPScript) 98
- Connection Line
  - 2D Curve 303
  - 3D Curve 305
  - Changing for a Curve 339
  - Turning On a Color Palette 340
- Constant (FPScript) 230
- Context Menu
  - Adding and Showing 480
  - Customizing 409
- Contour Lines
  - 3D Curve 305
  - Adding 335
  - Adding Labels 335
  - Editing 335
- Contour Lines in a 3D Curve 305
- Converting
  - Analysis Object or Data Link Object into Formulas 109
  - Formulas into Data Sets 109
- Coordinates
  - Inserting into a Diagram, Document or Worksheet 186
- Coordinates Window
  - (Description) 161
  - Adding Custom Coordinates 187
  - Changing Number Formatting 185
  - Copying a Value 185
  - Copying to Clipboard 186
  - Display Options 185
  - Displaying a Presentation Object 188
  - Displaying/Hiding 184
  - Displaying/Hiding Custom Coordinates 187
  - Editing Custom Coordinates 187
  - Opening a Data Set 185
- Creating
  - Document Template 389
  - Presentation Template 389
- Creating a List 363
- Cross-Hair Cursor 161
- Cursor
  - Accessing with FPScript 161
  - Adapting the Image Section to the Curve 173
  - Animate 161, 170
  - Bound 161
  - Canceling Zoom Actions 173
  - Changing the Number and Spacing of Harmonic Cursors 167
  - Changing Values 161
  - Compressing the Curve 172
  - Copying Data to the Clipboard 184
  - Dimensioning a Curve 189
  - Displaying Curve Data 331
  - Displaying the Curve Data 184
  - Displaying/Hiding 167
  - Editing Data 182
  - Excluding a Curve 167
  - Free 161
  - in Document 369
  - Interpolating Data 183
  - Markers 161
  - Moving Synchronously 171
  - Moving with the Mouse 168
  - Navigating to a frame in a Video 170
  - Navigating to a Value in a Diagram 170
  - Navigating to Markers 169
  - Navigating to Peak Values 169
  - Playback 161
  - Scrolling 172
  - Set Values To Void 161
  - Setting and Deleting Markers 174
  - Setting Values to Void 183
  - Stretching the Curve 172
  - Stretching the Image Section 171
  - Translating a Curve 189, 190
  - Type 161

- 
- Cursor
    - Using the Keyboard to Move 168
    - Using the Mouse to Zoom 172
    - Zoom 161
    - Zooming Automatically 174
    - Zooming Synchronously 171
  - Cursor Environment 161
  - Cursors
    - Adjusting Image Section when Moving Cursors 173
    - Toggling Harmonic 167
  - Curve
    - (Description) 302
    - 3D Columns (3D-Diagram) 305
    - Adding a Label 332
    - Adding a Perpendicular Line 343
    - Adding in 2D Diagram 330
    - Adding to Legend 345
    - Axis Label 310
    - Bar Chart (2D Diagram) 303
    - Changing Axis Assignment 330
    - Changing Bottoms for Columns/Bars 335
    - Changing Column/Bar Width 334
    - Changing Fill Color and Style 337
    - Changing Fill Direction 336
    - Changing Number Formatting for Data Labels 338
    - Changing the Connection Line 339
    - Changing the Symbol Shape for a Curve 344
    - Color Ramp 336
    - Column Chart (2D Diagram) 303
    - Connection Line (2D Diagram) 303
    - Contour (3D Diagram) 305
    - Curve Label 310
    - Data Labeling (2D Diagram) 303
    - Designing Columns/Bars 334
    - Determining the Interval Limits for a Color Palette 341
    - Differential Column Chart (2D Diagram) 303
    - Editing a Color Palette 342
    - Error Indicators (2D Diagram) 303
    - Excluding Curve from Autoscaling 330
    - Excluding from Autoscaling 330
    - Excluding from Cursoring 167
    - Handling Void Values 340
    - Hiding 331
    - Positioning the Label 332
    - Rearranging in Diagram 329
    - Removing from Legend 345
    - Resizing the Symbol for a Curve 344
    - Scrolling 170
    - Selecting Data to Label 338
    - Smoothing a Surface 333
    - Space Curve (3D Diagram) 305
    - Spline (2D Diagram) 303
    - Step Line 333
    - Step Line (2D Diagram) 303
    - Surface (3D Diagram) 305
    - Switching On a Display Format 329
    - Symbols (2D Diagram) 303
    - Tolerance Band (2D Diagram) 303
    - Turning On a Color Palette 340
    - Waterfall (3D Diagram) 305
  - Curve Diagram
    - 3D Curve Types and Display Formats 305
  - Curve Label
    - Positioning 332
  - Curve Type (3D Diagram) 305
  - Custom Coordinates
    - Adding 187
    - Displaying/Hiding 187
    - Editing 187
  - Custom Toolbar
    - Creating 408
    - Renaming 418
  - Customizable User Interface (Description) 403
  - Customizing
    - Decimal Symbol 425
    - Messages 425
    - Undo Function 425
-

- Customizing
  - Units 425
  - User Interface (Description) 403
- Customizing HTML Templates 71
- Customizing Messages 425
- Customizing the Decimal Symbol 425
- Customizing the Undo Function 425
- Customizing Units 425

## - D -

### Data

- Accessing (FPScript) 219
- Calculating with FPScript Functions 243
- Displaying in a Cell Table 359
- Displaying in Text 363
- Displaying the Curve under the Cursor 184
- Editing with Cursors 182
- Extracting with Cursors 184
- Formatting 393
- Importing Data Using Data Explorer 146
- Importing from Other Applications 143
- Importing Using DataQuery 148
- Interpolating with Cursors 183
- Scaling 243

### Data Analysis

- (Description) 191
- Efficiency 197
- Structuring 194

### Data Conversion

- AbsoluteTime Operator (FPScript) 254
- AbsoluteTimeStrict Operator (FPScript) 254
- RelativeTime Operator (FPScript) 256
- RelativeTimeStrict Operator (FPScript) 256

Data Exchange Format 148

### Data Explorer

- Change View 157
- Importing Data 146
- Search 158
- Test Series Analysis Tutorial 32
- Updating the View 158

Data Explorer (Option) 154

### Data Header

- of a Data Object 98

### Data Labeling

- 2D Curve 303
- 3D Curve 305
- Changing Number Formatting 338
- Changing Text Position and Direction 339
- Selecting Curve Data 338

### Data Link Object

- Converting into a Data Set 109
- Converting into a Formula 109
- Creating for OLE 145
- Editing Header Information 108
- Exporting 148
- Exporting as a File 153
- Exporting via OLE 153
- Exporting via the Clipboard 154
- OLE 138
- Playing as Audio File 107

Data Management 20

Data Matrix (FPScript) 98

### Data Object

- (Description) 98
- Assigned 98
- Component 98
- Data Order 98
- Data range lower limit 98
- Data range upper limit 98
- Data Structure 98
- Data Type 98
- Data Unit 98
- Exporting as a File 153
- Exporting via OLE 153
- Exporting via the Clipboard 154

- 
- Data Object
    - Header Information 98
    - Time Stamp 98
  - Data Order 98
  - Data Query
    - Concatenation 159
    - Setting Up 159
  - Data range lower limit 98
  - Data range upper limit 98
  - Data Series (FPScript) 98
  - Data Set
    - (Description) 110
    - Adding an X or Z component 114
    - Adding Columns and Rows 119
    - Adding Data from Clipboard 121
    - Adding Data from Object List 120
    - Calculating with an FPScript Function 243
    - Changing and Adding Values 117
    - Changing the Column Width 115
    - Changing the Data Type 114
    - Converting Formulas into Data Sets 109
    - Copying Data via Drag-and-Drop 121
    - Creating 113
    - Deleting Columns and Rows 120
    - Editing Header Information 108
    - Editing with Cursors 182
    - Exporting 148
    - Exporting as a File 153
    - Exporting via OLE 153
    - Exporting via the Clipboard 154
    - Merging 114
    - Opening 113
    - Opening in the Coordinates Window 185
    - Playing as Audio File 107
    - Selecting as a Component 114
    - Selecting Data 116
    - Using in VBA 459
  - Data Set Header Information
    - Displaying in Text 363
  - Data Set Information
    - Displaying in the Column Title 356
    - Displaying in the Legend 346
    - TimestampY Property (FPScript) 266
    - TimestampZ Property (FPScript) 267
  - Data Set Window
    - Adding Columns and Rows 119
    - Adding Data from Clipboard 121
    - Adding Data from Object List 120
    - Changing the Column Width 115
    - Deleting Columns and Rows 120
    - Navigate 116
    - Selecting Data 116
  - Data Sets
    - Arranging in Data View 113
    - Assigning to Each Other 110
  - Data Structure
    - (Description) 98
    - as Curve 302
    - Building 226
  - Data Type
    - (Description) 98
    - Changing 114
  - Data Unit 98
  - Data View
    - (Description) 110
    - Adding an X or Z component to Data Sets 114
    - Adding Columns and Rows 119
    - Adding Data from Clipboard 121
    - Adding Data from Object List 120
    - Arranging Data Sets 113
    - Assigning Data Sets 110
    - Calculating Values in Columns 122
    - Changing and Adding Values 117
    - Changing the Column Width 115
    - Copying Data via Drag-and-Drop 121
    - Creating a Data Set 113
    - Deleting Columns and Rows 120
    - Filling Columns with Values 122
    - Merging Data Sets to a Data Matrix 114
    - Navigate 116
-

- Data View
  - Selecting Data 116
  - Selecting Data Sets as Components 114
- Data Window
  - (Description) 110
- Database
  - Save Options 70
  - Updating Objects in the Background 91
- DataQuery
  - (Description) 158
  - Importing Data 148
  - Test Series Analysis Tutorial 32
- Date
  - Inserting Field in Text 363
- DCOM
  - (Description) 500
  - Configuring under Windows 2000 503
  - Configuring under Windows Vista 502
  - Configuring under Windows XP 502
  - Remarks 504
- DDE
  - (Description) 504
  - Connection Setup, Interaction and Termination 504
  - Visual Basic Example 504
- Debugger
  - Monitoring Variable Content 207
  - Setting and Removing Breakpoints 206
  - Starting 205
  - Stepping into Statements 206
- Debugger (FPScript) 204
- Debugging in Formulas 204
- Declaring Variables (FPScript) 225
- Defragmenting 64
- Developer Suite Edition 9
- Diagram
  - (2D Diagram Description) 297
  - 2D Display Formats 303
  - Adding a Boundary Line 189
  - Adding a Data Series 332
  - Adding a Dimension Line 189
  - Adding a Perpendicular 189
  - Adding a Symbol 189
  - Adding a Tangent 189
  - Adding a Tolerance Band 189
  - Adding or Removing an Axis 320
  - Adjusting the Height of Diagram Planes 321
  - Axis 298
  - Changing Lighting Parameters 319
  - Changing Page Splitting 318
  - Changing the Background Color 317
  - Changing the Viewing Angle 318
  - Color Legend 315
  - Color Palette and Color Gradient 314
  - Creating 315
  - Creating a Template 389
  - Cursor Environment 161
  - Designing Grid Elements 317
  - Dimensioning a Curve 189
  - Displaying in the Coordinates Window 188
  - Displaying the Curve Data 184
  - Embedding Graphics 319
  - Enabling or Disabling Automatic Margin Correction 316
  - Exporting 387
  - Grid 315
  - Inserting Coordinates 186
  - Legend 309
  - Lighting 310
  - Multi-page 294
  - Opening a Curve Data Set 331
  - Organizing a Template 390
  - Polar Graph 312
  - Resizing 316
  - Showing a Page 318
  - Specifying Output Unit for Axis 320
  - Template 294
  - Toggling Grid Lines 317
  - Translating a Curve 189, 190
  - Using Templates 389



- 
- Diagram
    - View 310
  - Differential Columns (2D Curve) 303
  - Digital Certificate
    - Obtaining 436
  - Digital Data
    - Displaying as a Step Line 333
  - Digital Signatures 430
  - Dimension (SI) 127
  - Dimension Line
    - Adding to a Diagram 189
  - Display Format
    - Bars (2D Diagram) 303
    - Border Lines (3D Diagram) 305
    - Columns (2D Diagram) 303
    - Connection Line (2D Diagram) 303
    - Connection Line (3D Diagram) 305
    - Contour Lines (3D Diagram) 305
    - Data Labeling (2D Diagram) 303
    - Data Labeling (3D Diagram) 305
    - Differential Columns (2D Diagram) 303
    - Error Indicators (2D Diagram) 303
    - Fill (2D Diagram) 303
    - Fill (3D Diagram) 305
    - Grid Lines (3D Diagram) 305
    - Perpendicular Lines (3D Diagram) 305
    - Projections (3D Diagram) 305
    - Switching On for a Curve 329
    - Symbols (2D Diagram) 303
    - Symbols (3D Diagram) 305
  - Displaying Hidden Lines 333
  - Displaying Text as Subscript 382
  - Displaying Text as Superscript 382
  - Displaying the Purpose of Linked Objects 85
  - Division
    - (Description) 298
    - Adding or Deleting a Label 323
    - Formatting the Label 324
    - Labeling 298
    - Labeling via Data Set 298
    - Labeling via Function 298
    - Labeling with a Unit 325
    - Labeling with Text 324
    - Positioning the Label 324
  - Document
    - (Description) 369
    - Adding Labels 379
    - Adding or Deleting Arrowheads 380
    - Adding/Deleting a Page 372
    - Aligning Drawing Objects 377
    - Changing a Fill Color or Pattern 381
    - Changing the Drawing Grid 373
    - Changing the View 373
    - Connecting Drawing Objects 380
    - Converting a Presentation Object into a Drawing 375
    - Creating 371
    - Creating a Group 378
    - Creating a Template 389
    - Cursor 369
    - Cursor Environment 161
    - Cutting Drawing Objects 380
    - Displaying Multiple Pages 373
    - Editing an Embedded Object 376
    - Editing Object Links 375
    - Embedding Graphics 375
    - Exporting 387
    - Grouping 369
    - Header and Footer 377
    - Inserting Coordinates 186
    - Linking and Embedding Presentation Objects 374
    - Linking/Embedding OLE Objects 376
    - Multi-page Diagrams and Column Tables 294
    - Organizing a Template 390
    - Page Layout 369
    - Template 294
    - Using Templates 389
    - Viewing a Page 372
-

### Document Template

- Creating 389
- Organizing 390
- Using 389

### Drawing

- Adding an Element 379
- Adding Labels 379
- Adding or Deleting Arrowheads 380
- Aligning Objects 377
- Changing a Fill Color or Pattern 381
- Changing Line and Fill Transparency 383
- Changing the Line Color 382
- Changing the Line Style or Border 382
- Connecting Objects 380
- Creating 378
- Creating a Group 378
- Cutting 380
- Grouping 369
- in Presentation Objects 294
- Rearranging on a Plane 377
- Repeating on Every Page 377
- Rotating 378
- Ungrouping 378

### Drop-Down List

- Resizing 413

### Dynamic Column Width 303

## - E -

### Embedded FPScript 294

### Embedding 294

### Embedding an Image in a Document 375

### Embedding Graphics in a Document 375

### Empty (FPScript) 98

### End-User License Agreement (DE AT CH) 507

### Error Indicators

- 2D Curve 303

### Changing the Total Number 337, 344

### Changing Type 337

### Resizing 337

### EULA (DE AT CH) 507

### EULA (International) 513

### Event

- Example (VBA) 495
- Using in VBA 446
- Using with Class Objects 468
- Using with FlexPro Objects 467
- Using with the Application or Databases Object 465
- Using with the Database Object 466

### Event Log

- (Description) 75
- Clearing 87
- Customizing 87
- Fixing the Cause of Events 87

### Example

- Analysis Automation 32
- Custom FPScript Function (VBA) 488
- Custom Import Filter (VBA) 481
- Data Management 20
- Displaying a Diagram on a Form (VBA) 493
- Displaying Data Graphically 28
- Documentation 29
- Editing Data 25
- Getting to Know FlexPro in just 15 Minutes 20
- Importing and Exporting Graphics 30
- Importing Data 25
- Processing Events (VBA) 495
- Selecting Data Ranges 27
- Signals 23
- Test Series Analysis 32
- Tips for New Users 20
- Using Data in Calculations 26
- Visually Analyzing Data 31

### Excel

- Analyzing and Presenting Data 126

**Excel**

- Creating a Workbook 123
- Creating Data Link Objects 126
- Data Link 123
- Exchanging Data with FlexPro 138
- Exporting a Workbook 124
- Importing a Workbook 124
- in FlexPro 122
- Manually Creating a Data Link 125
- Opening a Workbook 123
- Switching between External Storage and Embedding 124
- Tips for New Users 20
- Troubleshooting 124

**Excel Workbook 122****Exception Handling**

- (Description) 214

**Exporting**

- (Description) 148
- Data in File 148
- Data Objects as a File 153
- Data Objects via OLE 153
- Data via OLE 148
- Data via the Clipboard 154
- Excel Workbook 124
- Presentation Objects, Documents and Worksheets 387
- Presentation Objects, Documents and Worksheets with OLE 388

**Exporting OLE Objects 388****Extracting Data (FPScript) 217****- F -****Field**

- (Description) 399
- for Cell Table Data 359
- in Text 360
- Inserting a User Prompt into Text 364

- Inserting Data Set Header Information into Text 363
- Inserting Date or Time 363
- Inserting for Data in Text 363
- Post-Editing a Formatter 364

**File**

- Importing 141
- Importing Data Using Data Explorer 146
- Importing Using DataQuery 148

**File Indexing 154, 156**

- setting up 156

**Fill**

- 2D Curve 303
- 3D Curve 305
- Changing 381
- Changing Color and Style 337

**Finding**

- Objects in Project Database 54

**FlexPro**

- Data Exchange Format 148
- Developer Suite 9
- Professional 9
- Standard 9
- View 9

**FlexPro End-User License Agreement (International) 513****Floating Point Value (FPScript) 98****Floating Point Values**

- Changing Number of Fractional Digits for Output 426

**Floor 298****Folder**

- Displaying Active 83

**Folders**

- Displaying in the Object List 79
- Displaying Parent Folder 83
- Displaying Several 83

**Folders Window**

- (Description) 75

- Footer 377
  - Formatter
    - (Description) 393
    - Post-Editing the Field 364
  - Formatting a Paragraph 362
  - Formatting Numbers
    - Formatter 393
  - Formula
    - Calling an FScript Function 243
    - Converting Analysis Objects or Data Link Objects into Formulas 109
    - Converting into a Data Set 109
    - Creating 201
    - Creating an FScript Function 242
    - Debugger 204
    - Debugging 204
    - Editing 202
    - Editing Header Information 108
    - Exporting 148
    - Exporting as a File 153
    - Exporting via the Clipboard 154
    - Inserting Code Elements 202
    - Monitoring Variable Content 207
    - Playing as Audio File 107
    - Replacing in 204
    - Result 216
    - Saving FScript Functions as a Template 243
    - Searching in 204
    - Setting and Removing Breakpoints 206
    - Starting the Debugger 205
  - Formulas
    - (Description) 200
    - Exporting via OLE 153
    - Stepping into Statements 206
  - FPAccess Interface 498
  - FScript
    - (Description) 207
    - Arithmetical Expressions 208
    - Comments 222
    - Constant Quantities 230
    - Efficiency 197
    - Embedded 294
    - Exception 214
    - Functions 215, 241
    - Index 217
    - Percentage Calculation 209
    - Physical Quantity 209
    - Pre-Defined Constants 230
    - Quantity 209
    - Statements 222
    - Troubleshooting 214
    - Unit Calculation 209
    - Value Index 217
    - Variable 225
  - FScript Function
    - Calling 243
    - Creating 242
    - Saving as a Template 243
  - FScript Functions
    - Applying to Data Sets 243
  - Frame
    - Adding or Deleting for Legend 347
  - Free Cursor 161
  - Function
    - FScript 241
  - Functions (FScript) 215
- G -
- Graphics
    - Embedding in a Diagram 319
    - Embedding in a Table 351
  - Grid
    - Adding Projection 343
    - Designing Elements 317
    - Displaying Hidden Lines 333
    - for Diagrams 315
    - Toggling 317

Grid  
Turning On a Color Palette 340

Grouping  
Drawing Objects 369  
Ungroup 378

## - H -

Harmonic Cursors  
Changing the Number and Spacing 167  
Toggling 167

Header 377

Header Information  
(Description) 98  
Accessing (FPScript) 239  
Editing 108  
of a Data Object 98

Hierarchy View  
Copying Objects and Linked Objects 95  
Deleting an Object 96

HTML 71

HTML Export  
(Description) 71  
Complete Database 74  
Customizing HTML Templates 71  
Exporting Multiple Project Databases 74  
Individual Objects 74

Hyperlink  
(Description) 391  
Adding to a Diagram 392  
Assigning to an Object 392  
Following 392  
Inserting into a Document 391

## - I -

Image  
Embedding in a Diagram 319

Embedding in a Table 351

Image Marker  
(Description) 161  
Assigning Media 176  
Customizing the Display 180  
Formatting 179  
Positioning 176  
Setting and Deleting 174

Import  
Correcting Units during 127  
Graphics 375  
OLE Object 376  
Using Data Explorer 146

Importing  
(Description) 138  
Data from Other Applications 143  
Excel Workbook 124  
ODBC 145  
of Files 141  
Using DataQuery 148

Index  
(Description) 217

Index Column 347

Indexing 156  
Setting Up 71

Indicators  
2D Curve 303  
Changing the Total Number 337, 344  
Changing Type 337  
Resizing 337

Inserting a User Prompt into Text 364

Inserting Symbols in Text 382

Installation 11

Integer (FPScript) 98

ISO MME 138

Isometric Axis Adjustment 298  
Toggling On and Off 323

**- K -**

- Key Combination  
(Description) 403
- Keyword
  - As Object (FScript) 262
- Known Unit 127

**- L -**

- Label
  - Positioning for a Table 351
- Legend
  - (Description) 309
  - Adding a Curve 345
  - Adding a Title 346
  - Adding or Deleting 345
  - Adding or Deleting Frames 347
  - Changing the Arrangement of the Curves 346
  - Changing the Number of Columns 346
  - Displaying Data Set Information 346
  - Positioning 345
  - Removing a Curve 345
- License
  - Canceling Checkout 17
  - Offline Activation 15
  - Online Activation 15
  - transfer 17
  - Troubleshooting 18
- License Agreement (DE AT CH) 507
- License Agreement (International) 513
- License Manager
  - (Description) 13
  - Opening 14
- Lighting (3D Diagram) 310
- Lighting Parameters

- Changing for a 3D Diagram 319
- Line Cursor 161
- Link 294
- Linking/Embedding an OLE Object 376
- List
  - Data Type 98
- Local Indexing
  - Setting Up 71
- Logic
  - Logical Expressions (FScript) 234

**- M -**

- Macro
  - Auto Macros 471
  - Automating 428
  - Creating 432
  - Deleting 434
  - Editing 433
  - Recording to Generate Code 469
  - Revising 470
  - Running 434
  - Storing Values When Finished 472
  - Tips for Recording 433
  - Troubleshooting 438, 441
- Macro Project
  - Copying 435
  - Deleting 435
  - Digitally Signing 436
  - Renaming 435
- Macro Virus Protection
  - Changing Security Level 432
  - Digital Signatures 430
  - for Project Databases 429
  - Security Levels 431, 442
  - Troubleshooting 438
- Main Window 75
- Management
  - Data Management 20

**Marker**

- (Description) 161
- Accessing with FPScript 161
- Assigning Media 176
- Copying to Clipboard 182
- Customizing the Display 180
- Editing Size 178
- Editing Text 176
- Formatting 179
- Positioning 176
- Setting and Deleting 174

**Media 161**

- (Description) 364
- Adjusting the Volume 368
- Changing Playback Speed 368
- Choosing the Still Image 368
- Creating 366
- Creating the Time Base 367
- Playback 367

**Menu**

- (Description) 403
- Adding 404
- Animating 412
- Copying 405
- Deleting 411
- Grouping on Toolbar 408
- Moving 405, 409
- Renaming 410

**Menu Bar**

- Adding and Managing 474

**Menu Command**

- Adding 404
- Adding and Managing 474
- Adding Image 417
- Adding Text 417
- Deleting 405
- Moving 405
- Renaming 410

**Metadata**

- of a Data Object 98

**Method**

- (VBA Description) 443

**MME (TS 13499) 138****Mouse Wheel**

- Scrolling in a Diagram 172

**Multichannel Analysis 244****Multi-page Diagrams and Column Tables 294****- N -****Naming Conventions**

- for FlexPro Objects 87

**Naming Conventions for Objects 87****Navigating to Markers 169****Navigating to Peak Values 169****Network License**

- Canceling Checkout 17
- Checking Out 16
- License Manager 13
- transfer 17

**Number Format**

- Changing for Cell Table 360
- Changing for Text 362
- Changing for the Column Table 354
- Changing for the Coordinates Window 185

**Number Formatting**

- Changing for Curve Data Labels 338

**- O -****Object**

- (Description) 87
- (VBA Description) 443
- Copying 94
- Copying with Linked Objects 95
- Creating in a Project Database 90

**Object**

- Deleting 96
- Displaying the Properties Dialog Box 93
- Editing Parameters 96
- Finding in Project Database 54
- Linking/Embedding 376
- Lock 87
- Locking 92
- Moving 94
- Name Extensions 87
- Naming Conventions 87
- Opening 93
- Parameters 87
- Printing 97
- Read-Only 87
- Renaming 93
- Selecting 80
- Selecting with Linked Objects 85
- Undo/Redo 93
- Updating 87, 90
- Updating in the Background 91
- Write Protecting 92

**Object Hierarchy**

- (Description) 75, 453
- (Overview) 455
- Displaying Linked Objects 84
- Displaying the Purpose of Linked Objects 85
- Selecting an Object and Linked Objects 85

**Object List**

- (Description) 75
- Adding an X or Z component to Data Sets 114
- Adding/Removing Columns 84
- Categorizing Objects 92
- Changing the Column Width 82
- Copying Objects 94
- Copying Objects and Linked Objects 95
- Creating Objects 90
- Deleting Objects 96

**Displaying an Object's Properties Dialog Box** 93

- Displaying Folders 79
- Displaying Object Categories 82
- Displaying Objects in Groups 84
- Displaying Parent Folder 83
- Displaying the Active Folder 83
- Locking Objects 92
- Manually Sorting 81
- Merging Data Sets to a Data Matrix 114
- Moving Objects 94
- Opening an Object 93
- Renaming an Object 93
- Re-Sorting Columns 84
- Selecting an Object 80
- Selecting the Display Type 80
- Sorting 81
- Undo/Redo 93
- Updating Objects 90
- Updating Objects in the Background 91
- Write Protecting Objects 92

**Object Model** 452**Object Properties**

- Accessing (FPScript) 218

**Object Reference** 219**Object Reference (FPScript)** 98**Objects**

- Categorizing 92

**ODBC**

- Importing 145
- Importing Data (Description) 138

**ODS**

- Importing Data (Description) 138

**OLE**

- Creating a Data Link Object 145
- Data Export 153
- Data Link 138
- Linking and Embedding 294
- Programmatic Identifiers 464

**Operator**



**Operator**

- AbsoluteTime (FPScript) 254
- AbsoluteTimeStrict (FPScript) 254
- Associativity (FPScript) 234
- Description (FPScript) 234
- Precedence (FPScript) 234
- RelativeTime (FPScript) 256
- RelativeTimeStrict (FPScript) 256

**Optimization**

- Data Analysis 197

**Optimizing**

- for Large Volumes of Data 64

**Optimizing Data Throughput 64****Option**

- Data Explorer 154

**Organize**

- Analysis Templates 198, 199

**Organizer**

- Units 137

**Organizing**

- Document Templates 390
- Presentation Templates 390

**Orientation Arrow**

- Adding or Deleting 325
- Positioning 325

**Output Format**

- for Complex Numbers 426
- for Floating Point Values 426

**Output Unit**

- Column Table 347
- Diagram Axis 297
- Specifying for Axis 320
- Specifying for Table Cell 358
- Specifying for Table Column 352

**- P -****Page**

- Adding/Deleting 372
- Inserting Numbers 372
- Setting Up Format 372
- Showing a 2D Diagram Page 318

**Page Layout**

- Document 369

**Page Splitting (2D Diagram) 318****Palette**

- Determining the Interval Limits for a Curve 341
- Editing for a Curve 342
- Transferring 342
- Turning On 340

**Parameters**

- Editing for an Object 96
- for Object 87

**Path Name 219****Percentage Calculation (FPScript) 209****Perpendicular**

- Adding to a Diagram 189

**Perpendicular Line**

- 3D Curve 305
- Adding 343

**Perpendicular Lines**

- Editing 343

**Perspectives**

- Changing for a 3D Diagram 318

**Physical Quantity**

- Calculating and Converting 127
- Physical Quantity (Description) 127

**Physical Quantity (FPScript) 209****Placeholder 399****Plane**

- Color Ramp 336

**Play**

- Audio Signal 367
- Media 367
- Video 367

- Play Speed
  - Changing for Media 368
- Playback
  - Adjusting the Volume 368
  - Audio Signal 367
  - Media 367
  - Signal (Cursor Environment) 161
  - Video 367
- Playback Speed
  - Changing for Media 368
- Playing 107
- Polar Graph (2D Diagram) 312
- Position
  - Color Legend 318
  - Legend 345
- Precedence of Operators (FPScript) 234
- Presentation Object
  - (Description) 294
  - 2D Diagram 297
  - 3D Diagram 297
  - Cell Table 347
  - Column Table 347
  - Convert into Drawing 375
  - Editing Links 375
  - Exporting 387
  - Media 364
  - Template 294
  - Text 360
- Presentation Objects
  - Linking and Embedding 374
- Presentation Template 294
  - Creating 389
  - Organizing 390
  - Using 389
- Preview
  - (Description) 75
- Printing 97
- Professional Edition 9
- Programmatic OLE Identifiers 464
- Project Database
  - (Description) 50
  - Categorizing Objects 92
  - Compatibility 50
  - Competing Access 50
  - Copying Objects 94
  - Copying Objects and Linked Objects 95
  - Creating 53
  - Creating Objects 90
  - Defragmenting 64
  - Deleting Objects 96
  - Displaying an Object's Properties Dialog Box 93
  - Displaying Object Categories 82
  - Displaying Parent Folder 83
  - Displaying Several Folders 83
  - Finding Objects 54
  - Locking Objects 92
  - Modes for Opening 50
  - Moving Objects 94
  - Opening an Object 93
  - Printing Objects 97
  - Renaming an Object 93
  - Save Options 50
  - Selecting an Object 80
  - Sending via E-mail 64
  - Setting Up Indexing 71
  - Storage Formats 50
  - Troubleshooting 68
  - Undo/Redo 93
  - Updating Objects 90
  - Using in VBA 455
  - Write Protecting Objects 92
- Project Databases
  - Exporting 71
- Projections
  - 3D Curve 305
  - Adding for a Space Curve Diagram 343
  - Adding for a Surface Chart 343
  - Adding for Grid Display 343
- Properties

**Properties**  
     Accessing (FPScript) 218  
     Displaying in Groups 86  
     Editing 85  
     Show/Hide with Read-Only Access 86  
     Showing/Hiding a Group 86  
     Showing/Hiding Items That are not Shared 86  
     Sorting 86  
**Properties Window (Description)** 75  
**Property**  
     (VBA Description) 443  
     TimestampY (FPScript) 266  
     TimestampZ (FPScript) 267

## - Q -

**Quantity**  
     Calculating and Converting 127  
     Physical Quantity (Description) 127  
**Quantity (FPScript)** 209

## - R -

**Range Marker**  
     (Description) 161  
     Customizing the Display 180  
     Setting and Deleting 174  
**Range of Data (FPScript)** 217  
**Rearranging**  
     Cell Table Columns or Rows 358  
     Columns in a Column Table 353  
**Rearranging on a Plane** 377  
**Recording**  
     Macro 432  
**Redo**  
     for an Object in the Object List 93  
**Relative Path Name** 219

**Relative Time** 98  
     RelativeTime Operator (FPScript) 256  
     RelativeTimeStrict Operator (FPScript) 256  
**RelativeTime Operator (FPScript)** 256  
**RelativeTimeStrict Operator (FPScript)** 256  
**Replacing**  
     in a Formula 204  
**Reusing Display Attributes** 381  
**Roaming User Profile** 424  
**Run Mode (Formula Debugger)** 204  
**Runtime Error** 204

## - S -

**Scalar Value (FPScript)** 98  
**Scaling**  
     Across Data Range End Values 298  
     Automatic 298  
     Metric 298  
     of Data 243  
**Scaling Types** 298  
**ScreenTip**  
     Showing or Hiding Shortcuts 419  
**ScreenTips**  
     Showing or Hiding 419  
**Scrolling** 172  
     Curve 170  
**Search Filters**  
     Selecting 158  
     Setting Up 158  
**Search Item** 54  
**Search Operation** 54  
**Search Results**  
     (Description) 75  
**Search Value** 54

- Searching
  - for Indexed Data 158
  - for Text 361
  - in a Formula 204
  - in Data Explorer 158
- Security Level
  - Changing for Macro Virus Protection 432
- Security Levels 442
- Semantic Error 204
- Setting Tabs in Paragraphs 362
- Setting Up
  - File Indexing 156
- Shortcut
  - (Description) 403
  - Restoring 420
  - Showing or Hiding in ScreenTips 419
- Shortcut key
  - Assigning 407, 420
  - Changing 407
  - Removing 420
  - Restoring 420
- SI
  - Base Units 127
  - Dimension 127
  - Unit System (Description) 127
- Signal
  - Data Structure (FPScript) 98
- Signal Series (FPScript) 98
- Single Step Mode (Formula Debugger) 204
- Single User License
  - License Manager 13
  - Offline Activation 15
  - Online Activation 15
  - transfer 17
  - Troubleshooting 18
- Smart Update 191
- Sorting
  - Curves in a Diagram 329
- Space Curve
  - (FPScript) 98
  - 3D Curve 305
  - Adding a Perpendicular Line 343
  - Adding Projection 343
  - Color Ramp 336
  - Editing a Perpendicular Line 343
  - Turning On a Color Palette 340
- Splitting
  - FPScript Statements 222
- Standard Edition 9
- Standard Formatter 393
- Startup Switches for FlexPro 424
- Statement (Description) 222
- Step Into (Formula Debugger) 206
- Step Line
  - 3D Curve 305
- Step Line (2D Curve) 303
- Step Over (Formula Debugger) 206
- Still Image
  - (Media, Description) 364
  - Choosing for Media 368
- String
  - Formatting 393
- String (FPScript) 98
- Subdivisions
  - Changing the Number on the Axis 327
- Subfolder
  - Activating and Deactivating 83
- Submenu
  - Moving 409
- Surface
  - 3D Curve 305
  - Adding a Data Series 332
  - Adding Contour Line Labels 335
  - Adding Contour Lines 335
  - Adding Projection 343

**Surface**

- Changing Fill Color and Style 337
- Changing the Grid Lines 339
- Color Ramp 336
- Determining the Interval Limits for a Color Palette 341
- Displaying Hidden Lines 333
- Editing a Color Palette 342
- Editing Contour Lines 335
- Smoothing 333

**Symbol**

- Adding to a Diagram 189

**Symbols**

- Changing the Shape in a Curve 344
- Resizing in a Curve 344
- Turning On a Color Palette for a Space Curve Diagram 340

**Syntax Error 204****- T -****Table**

- (Description) 347
- Adding a Column 357
- Adding a Column Title 355
- Adding a Data Column 352
- Adding a Row 357
- Adding an Index Column 352
- Adding or Deleting a Column Index 356
- Adding or Deleting a Label 351
- Adding or Deleting Dividing Lines and Frames 350
- Alternating Colors 354
- Changing Colors 350
- Changing the Column Width 353
- Changing the Font 350
- Changing the Row Height 353, 359
- Changing the Text Alignment 351
- Creating 349
- Creating a Template 389

**Displaying in the Coordinates Window 188**

- Editing Custom Units 135
- Editing for Unit Correction during Data Import 136
- Embedding Graphics 351
- Exporting 387
- Multi-page 294
- Organizing a Template 390
- Outputting Data with a Unit Symbol 355
- Positioning the Label 351
- Rearranging Columns 353
- Show Page 354
- Specifying the Output Unit 352, 358
- Template 294
- Using Templates 389

**Tangent**

- Adding to a Diagram 189

**Template**

- Creating (Analysis Template) 198
- Creating (Document Template) 389
- Creating (Presentation Template) 389
- FPScript Function 243
- Organizing (Analysis Template) 199
- Organizing (Document Template) 390
- Organizing (Presentation Template) 390
- Using (Analysis Template) 198
- Using (Document Template) 389
- Using (Presentation Template) 389

**Template Database**

- Setting Up 70

**Test Series Analysis (Tutorial) 32****Test Series Analysis Tutorial 9****Text**

- (Description) 360
- Changing Number Format 362
- Copying Passages 361
- Creating 360
- Creating a List 363
- Creating a Template 389
- Deleting Passages 361

## Text

- Displaying in the Coordinates Window 188
- Edit 361
- Formatting Paragraphs 362
- Inserting a Field for Data Set Header Information 363
- Inserting a Field for Values 363
- Inserting Date or Time 363
- Organizing a Template 390
- Post-Editing a Formatter 364
- Searching and Replacing Text 362
- Searching for Text 361
- Setting Tabs 362
- Using Templates 389

## Text Data

- Importing 141

## Text Marker

- (Description) 161
- Customizing the Display 180
- Editing 176
- Editing Size 178
- Formatting 179
- Positioning 176

this 202

## Time

- Inserting Field in Text 363

## Time Base

- (Media, Description) 364
- Creating for Media 367

Time Span (FPScript) 98

Time Stamp 98

## Timespan

- Formatting 393

TimestampY Property (FPScript) 266

TimestampZ Property (FPScript) 267

Tips for New Users (Tutorial) 20

## Tolerance Band

- Adding to a Diagram 189

Tolerance Band (2D Curve) 303

## Toolbar

- (Description) 403
- Adding and Changing 476
- Changing Images 414
- Creating Custom 408
- Deleting 411
- Grouping Buttons 408
- Hiding 412
- Moving 412
- Renaming 418
- Resizing 413
- Resizing Buttons 413
- Restoring 418
- Showing 412
- Showing All Buttons 412
- Showing or Hiding ScreenTips 419

Transfer (License) 17

## Troubleshooting

- Single User License 18

Troubleshooting (FPScript) 214

Trusted Publishers 431

TS 13499 MME 138

## Tutorial

- Getting to Know FlexPro in just 15 Minutes 20
- Test Series Analysis 32
- Tips for New Users 20

# - U -

## Undo

- for an Object in the Object List 93

## Unit

- Calculating with (FPScript) 209
- Checking Imported 134, 137
- Column Table 347
- Corrections during Data Import 127
- Data Unit 98

**Unit**

- Diagram Axis 297
- Editing Tables for Unit Correction 136
- in Header Information 98
- Internal Representation 127
- Known and Unknown 127
- Notation (FPScript) 209
- Organizer 137
- Outputting Data to a Column Table with a Unit Symbol 355
- Registering Custom 135
- Setting Up the Unit Manager 133
- SI Base Units 127
- SI Dimension 127
- SI Unit System 127
- Specifying for Axis 320
- Specifying for Table Cell 358
- Specifying for Table Column 352
- Unit Collection 127
- Unit Manager (Description) 127

Unit Calculation (FPScript) 209

Unit Collection 127

Unit Manager  
(Description) 127  
Modes 127  
Setup 133

Unit System (SI) 127

Unknown Unit 127

Update  
Smart Update 191

Updating  
Views in Data Explorer 158

User Interface; Customizable (Description)  
403

User Profile  
Creating for User 422  
Installing for User 423  
roaming 424

User Prompt  
in Fields 399

**User Rights**

- Assigning 421
- Transferring 421

**Using**

- Analysis Template 198
- Document Template 389
- Presentation Template 389

**- V -**

Value Index (FPScript) 217

Variable (FPScript)  
Declaring 225  
Monitoring 207

Vector (2D Curve) 303

**Video**

- Adjusting the Volume 368
- as Presentation Object 364
- Changing Playback Speed 368
- Choosing the Still Image 368
- Creating the Time Base 367
- Playback 367

**View**

- Changing for Data Explorer 157

View (3D Diagram) 310

View Edition 9

Viewing Angle  
Changing for a 3D Diagram 318

Virus Protection 430

**Visual Basic**

- Programmatic OLE Identifiers 464
- Tips and Tricks 449

**Void Value**

- (Description) 229
- Displaying in a Curve 340
- Setting with a Cursor 183

**Volume**

- Adjusting for Media 368

**- W -**

Watch Window 204

## Waterfall

- 3D Curve 305
- Adding a Data Series 332
- Color Ramp 336
- Determining the Interval Limits for a Color Palette 341
- Editing a Color Palette 342
- Rearranging Curves 329
- Turning On a Color Palette 340

## Widow

- Resetting Arrangement 79

## Window

- Activating 78
- ASAM ODS Data Source (Description) 75
- Closing 78
- Event Log (Description) 75
- Folders (Description) 75
- Hiding 78
- Main (Description) 75
- Moving 78
- Object Hierarchy (Description) 75
- Object List (Description) 75
- Preview (Description) 75
- Properties (Description) 75
- Saving Workspace 64
- Search Results (Description) 75
- Showing 78

## Worksheet

- (Description) 383
- Adding or Removing Panes 386
- Changing the Window Layout 387
- Creating 384
- Creating for a Data Set with with a Broad Overview 385
- Cursor Environment 161
- Exporting 387
- Inserting Coordinates 186

Linking and Embedding 385

Maximizing a Pane 387

## Workspace

- Saving 64
- Saving Automatically 64

**- Z -**

## Zoom

- (Description) 161

## Zooming

- Range Between Cursors 171
- Synchronizing 171





The company and product names listed herein are registered trademarks of the respective companies.

FlexPro is protected by international copyright law. Copyright © 1991 – 2015 Weisang GmbH. Portions Copyright © 1996 Microsoft Corporation.

Dated 1/1/2015, subject to errors and changes.